

IHM4



28-31 Octobre 2014

Villeneuve d'Ascq, France

Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine



The Association for Computing Machinery

2 Penn Plaza, Suite 701
New York, New York 10121-0701

ACM COPYRIGHT NOTICE

Copyright © 2014 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept., ACM, Inc., fax +1 (212) 869-0481, or permissions@acm.org.

For other copying of articles that carry a code at the bottom of the first or last page, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, +1-978-750-8400, +1-978-750-4470 (fax).

NOTICE TO PAST AUTHORS OF ACM-PUBLISHED ARTICLES

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that was previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

WHEN CITING ARTICLES FROM THESE PROCEEDINGS, PLEASE USE THE FOLLOWING FORMAT:

Y. Gauthier, J. Marcy, D. Duprat, A. Paoleschi, C. Letondal, R. Lesbordes, J.L. Vinot et C. Hurter. "Interactions gestuelles pour Strip'TIC, un environnement tangible pour les contrôleurs aériens". In *Proceedings of IHM'14*, 26^{ième} Conférence de l'Association Francophone d'Interaction Homme-Machine (Villeneuve d'Ascq, France, 28-31 octobre 2014), p. 18-27. ACM, 2014.

ACM ISBN: 978-1-4503-2935-4



Association Francophone d'Interaction Homme-Machine

La Conférence Francophone sur l'Interaction Homme-Machine (IHM) est organisée annuellement par l'AFIHM, l'Association Francophone d'Interaction Homme-Machine.

L'AFIHM a pour but principal de promouvoir le savoir et les connaissances du domaine de l'Interaction Homme-Machine et des divers domaines concourant au savoir et aux connaissances facilitant la conception, la réalisation et l'évaluation des systèmes interactifs actuels ou futurs.

L'AFIHM offre aux chercheurs et praticiens de l'IHM un lieu d'échange et de savoir sur leur domaine professionnel via :

- l'organisation et le soutien de manifestations scientifiques : conférence IHM et autres conférences francophones, forums thématiques, groupes de travail, écoles d'été et rencontres jeunes chercheurs ;
- la publication du Journal d'Interaction Personne-Système et l'édition des actes de conférences ;
- des liens et des collaborations avec diverses organisations françaises et internationales.

AFIHM
Laboratoire LIG
Équipe I1HM
BP 53
38041 Grenoble Cedex 9
France

Contact : secretaire@afihm.org
Site Web : <http://www.afihm.org/>

Numéro de Siret : 410 093 892 00017

Éditions précédentes d'IHM

- 25 **IHM 2013**, Conférence, Bordeaux, France
- 24 **Ergo'IHM 2012**, Conférence IHM/ERGO-IA, Biarritz, France
- 23 **IHM 2011**, Conférence, Sophia Antipolis, France
- 22 **IHM 2010**, Conférence, Luxembourg
- 21 **IHM 2009**, Conférence, Grenoble, France
- 20 **IHM 2008**, Conférence, Metz, France
- 19 **IHM 2007**, Conférence, Paris, France
- 18 **IHM 2006**, Conférence, Montréal, Québec, Canada
- 17 **IHM 2005**, Conférence, Toulouse, France
- 16 **IHM 2004**, Conférence, Namur, Belgique
- 15 **IHM 2003**, Conférence, Caen, France
- 14 **IHM 2002**, Conférence, Poitiers, France
- 13 **IHM-HCI 2001**, Conférence conjointe de l'AFIHM et du British HCI Group, Lille, France
- 12 **Ergo-IHM 2000**, Conférence IHM/ERGO-IA, Biarritz, France
- 11 **IHM'99**, Conférence, Montpellier, France
- 10 **IHM'98**, Ateliers, Nantes, France
- 9 **IHM'97**, Conférence, Poitiers, France
- 8 **IHM'96**, Ateliers, Grenoble, France
- 7 **IHM'95**, Conférence, Toulouse, France
- 6 **IHM'94**, Ateliers, Lille, France
- 5 **IHM'93**, Ateliers, Paris, France
- 4 **IHM'92**, Ateliers, Lyon, France
- 3 **IHM'91**, Ateliers, Dourdan, France
- 2 **IHM'90**, Conférence couplée avec ERGO-IA, Biarritz, France
- 1 **IHM'89**, Conférence, Sophia Antipolis, France

Avant-propos

Interaction Homme-Machine, de retour dans le Nord !

L'Association Francophone d'Interaction Homme-Machine (AFIHM), Inria et l'Université Lille 1 ont le plaisir de présenter la 26^e édition de la conférence IHM. Il s'agit de la troisième édition lilloise, après 1994 et 2001.

IHM14 rassemble cette année la communauté IHM francophone non seulement autour de présentations d'articles scientifiques, d'ateliers, de démonstrations, mais aussi de Travaux en Cours (TeC). IHM14 inaugure cette nouvelle catégorie de soumissions, inspirée des Work in Progress de la conférence CHI, qui permet de présenter des travaux non achevés mais qui gagnent à être discutés avec la communauté. Ils sont présentés à la conférence sous forme de posters et sont inclus dans l'ACM digital library et HAL, au même titre que les articles scientifiques.

Les articles scientifiques présentent les résultats de travaux aboutis sur des aspects fondamentaux ou appliqués de l'IHM. Cette année, nous avons reçu 32 soumissions dans cette catégorie et nous avons sélectionné 18 articles qui seront présentés dans cinq sessions. Chaque article a été évalué par deux membres du comité de programme et deux relecteurs externes.

Les Travaux en cours (TeC) ont pour but de présenter de nouveaux travaux ou des travaux en cours. Nous avons reçu 19 soumissions dans cette catégorie et sélectionné 10 TeC qui seront présentés sous la forme d'une présentation éclairée et d'un poster durant des sessions dédiées. Chaque TeC a été évalué par l'un des responsables des TeC et 2 ou 3 relecteurs externes.

Nous tenons à remercier vivement toutes les personnes ayant participé à l'organisation d'IHM14, tant sur le plan scientifique que logistique ou financier. Nous remercions les auteurs, relecteurs et participants qui font vivre la communauté.

Géry Casiez et Thomas Pietrzak,
Présidents de la conférence

Olivier Chapuis et Stéphane Conversy,
Présidents du comité de programme

Comités de la conférence

Présidents de la conférence

Géry Casiez, Université Lille 1
Thomas Pietrzak, Université Lille 1

Soutien à l'organisation

Marie-Bénédicte Dernoncourt, Inria Lille

Comité local d'organisation

Jonathan Aceituno, Inria Lille
Fanny Chevalier, Inria Lille
Christophe Chaillou, Université Lille 1
Alix Goguey, Inria Lille
Xavier Le Pallec, Université Lille 1
Damien Marchal, CNRS & Université Lille 1
José Rouillard, Université Lille 1
Nicolas Roussel, Inria Lille

Présidents du comité de programme

Olivier Chapuis, CNRS & Univ. Paris-Sud
Stéphane Conversy, ENAC & Univ. de Toulouse

Travaux en Cours (TeC)

José Rouillard, Université Lille 1
Nadine Vigouroux, Univ. Paul Sabatier, Toulouse

Démonstrations

Damien Marchal, CNRS & Université Lille 1
Céline Coutrix, CNRS, Laboratoire
d'Informatique de Grenoble

Relations industrielles & sponsoring

Christophe Chaillou, Université Lille 1
Nadine Couture, ESTIA

Ateliers Groupes de Travail

Fanny Chevalier, Inria Lille
Marcos Serrano, Université Toulouse 3

Rencontres doctorales

Xavier Le Pallec, Université Lille 1
Monique Noirhomme, Université de Namur

Responsable communication

Nicolas Roussel, Inria Lille

Edition des actes

Alix Goguey, Inria Lille

Webmaster

Jonathan Aceituno, Inria Lille

Comité de programme

Gilles Bailly, CNRS, Télécom ParisTech, Paris
Nadine Couture, ESTIA
Sophie Dupuy-Chessa, UPMF, Grenoble
James Eagan, Télécom ParisTech, Paris
Nicolas Ferey, Université Paris-Sud – LIMSI
Patrick Girard, Université de Poitiers
Mountaz Hascoët, Université Montpellier II
Christophe Hurter, ENAC, Toulouse
Denis Lalanne, Université de Fribourg
Catherine Letondal, ENAC, Toulouse
Sylvain Malacria, UCL, Londres
Yannick Prié, Université de Nantes
Gaétan Rey, Université de Nice
Dominique Scapin, Inria, Rocquencourt
Audrey Serna, Université Claude Bernard, Lyon
Marcos Serrano, Université Toulouse 3
Jean-Claude Tarby, Université Lille 1
Theophanis Tsandilas, Inria, Orsay

Avec le soutien de...



LA COMMUNAUTÉ DE L'IMAGE
EN NORD-PAS DE CALAIS



Conférenciers invités



Kris Luyten

A Secret Weapon to Fight an Army of Underhand Machines

Abstract: The so-called disappearing computer dictates that hardware and software blends — even hides — within the natural environment of its user. The user often interacts with such underhand computer systems through sensors with which the system observes and analyses the user. This makes using the system transparent and apparently automatic, and lets the users believe the environment is behaving in an intelligent way. However, because these systems tend to act without explicitly involving the user, users might be surprised as to why their environment behaves in a certain way. Moreover, context-aware systems will sometimes make mistakes because of the inevitable incompleteness of information. In this talk we'll discuss how we should take the power back and demand machines that can account for what they do and how they do it. One key approach — our secret weapon — is to make an environment intelligible, which means the system tries to convey its current state and inner workings to the end-user — preferably when the end-user expresses the wish to be informed and to take control.

Bio: Kris Luyten is an associate professor in Computer Science at Hasselt University and member of the Human-Computer Interaction lab of the iMinds research institute Expertise Centre for Digital Media. His original, and still ongoing, research interest is finding new techniques and methods to engineer and use context-aware interactive systems. Since a few years, he is working within and with his team on various aspects of creating more accessible, usable and approachable ubiquitous systems. One keyword that comes up very often while doing this research, is intelligibility in its various shapes and forms. He thinks writing a bio in third person is a bit weird and likes to point to his website for more information on his endeavours in research and teaching: www.krisluyten.net

Conférenciers invités



Nicolas Nova

Science-Fiction, interfaces et design fiction

Résumé: Pour les chercheurs en IHM, les technologies présentées dans les oeuvres de science-fiction présentent un attrait particulier. Elles fournissent en quelque sorte une représentation des imaginaires techniques potentiels, tout en étant influencées par les craintes et les espoirs de l'époque qui les voient naître. Cette présentation abordera les relations mutuelles entre représentations des technologies du futur dans la science-fiction et le design d'interfaces. En s'appuyant sur divers projets de "design fiction" menés au Near Future Laboratory, elle montrera le rôle du champ du design et des spéculations quant au futur des technologies numériques. En particulier, la présentation s'intéressera à la façon dont ces projets produisent des visions alternatives à celles des grandes sociétés technologiques; en particulier en se penchant sur les échecs et les frictions causées par les nouvelles interfaces en cours de développement.

Bio: Nicolas Nova est co-fondateur du Near Future Laboratory et professeur à la HEAD-Genève (Haute-Ecole d'Art et de Design), où il enseigne l'ethnographie, l'histoire des cultures numériques et la recherche en design. Il est titulaire d'un doctorat en interaction homme-machine de l'EPFL (Lausanne, Suisse) et a été chercheur invité à Art Center College of Design (Pasadena, California). Auteur de "Joypads! Le design des manettes" (Les Moutons Electriques) avec L. Bolli et "Curious Rituals: Gestural Interactions in the Digital Everyday" (Near Future Laboratory Press), il est également responsable éditorial de Lift, conférence internationale sur l'innovation et les usages des technologies. Il s'intéresse à la recherche en design, aux questions d'usage et de prospective en lien avec les technologies numériques.

Relecteurs

Sonia Adelé
Leonardo Angelini
Jean-Yves Antoine
Olivier Beaudoux
Mireille Bétrancourt
Alan Blackwell
Renaud Blanch
Arnaud Blouin
Colin Blättler
Marc-Eric Bobillier Chaumon
David Bonnet
Gaëtan Bourmaud
Guy Boy
Anke Brock
Sybille Caffiau
Maurizio Caon
Loïc Caroux
Julien Castet
Mickael Causse
Selem Charfi
Fanny Chevalier
Aurélie Cohé
Bertrand David
Emmanuel Dubois
Bruno Dumas

Cédric Dumas
Thierry Duval
Jérémie Garcia
Nadia Ghaibi
Alain Giboin
Patrick Girard
Christophe Kolski
Sebastien Kubicki
François Lemieux
Nicole Lompre
Fabien Lotte
Marc Macé
Sylvain Malacria
Odile Martial
Christine Michel
Gabriel Michel
Mathieu Nancel
David Navarre
Luciana Nedel
Monique Noirhomme
René Patesson
Catherine Pelachaud
Charles Perin
Simon Perrault
Clément Pillias

Patricia Plénacoste
Mathieu Raynal
Helmuth Reiter
Philippe Renevier
Patrick Reuter
Meriem Riahi
Sophie Robert
Eve Ross
Anne Roudaut
Nicolas Roussel
Stéphane Safin
Imad Saleh
Pascal Salembier
Lepreux Sophie
Philippe Suignard
Franck Tarpin-Bernard
Daniela Trevisan
Philippe Truillet
Tuomas Vaittinen
Frédéric Vella
Jean-Marc Vezien
Jean-Luc Vinot
Romain Vuillemot

Table des matières

Session 1 : Systèmes mixtes

PaperComposer: Creating Interactive Paper Interfaces for Music Composition	1
Jérémie Garcia, Theophanis Tsandilas, Carlos Agon, Wendy Mackay	
A design space of guidance techniques for large and dense physical environments	9
Hind Gacem, Gilles Bailly, James Eagan, Eric Lecolinet	
Interactions gestuelles pour Strip’TIC, un environnement tangible pour les contrôleurs aériens	18
Yoann Gauthier, Joran Marcy, David Duprat, Alexis Paoleschi, Catherine Letondal, Rémi Lesbordes, Jean-luc Vinot, Christophe Hurter	

Session 2 : Techniques d’Interaction : Commandes et Gestes

Adoiraccourcix : Sélection de Commandes sur Écrans Tactiles Multi-Points par Identification des Doigts	28
Alix Goguey, Géry Casiez, Thomas Pietrzak, Daniel Vogel, Nicolas Roussel	
CtrlMouse et TouchCtrl : Dupliquer les Délimiteurs de Mode sur la Souris	38
Thomas Pietrzak, Sylvain Malacria, Gilles Bailly	
Manipulation de Sliders Multiples par Franchissement	48
Charles Perin, Pierre Dragicevic	
The Hotkey Palette: Flexible Contextual Retrieval of Chosen Documents and Windows	55
Jonathan Aceituno, Nicolas Roussel	

Session 3 : Systèmes Critiques

L'accident du vol AF447 Rio-Paris, un cas d'étude pour la recherche en IHM	60
Stéphane Conversy, Stéphane Chatty, Hélène Gaspard-Boulinc, Jean-luc Vinot	
Exigences d'utilisabilité pour les outils d'ingénierie des exigences	70
Hélène Gaspard-Boulinc, Stéphane Conversy	

Résilience des systèmes interactifs: contribution par une architecture tolérante aux fautes	80
Camille Fayollas, Philippe Palanque, Jean-charles Fabre, David Navarre, Eric Barboni, Martin Cronel, Yannick Deleris	

Session 4 : Techniques d'Interaction : Dimensions > 2

Conception d'un Dispositif pour Interagir avec des Données Multidimensionnelles : Disco	91
Gary Perelman, Marcos Serrano, Mathieu Raynal, Célia Picard, Mustapha Derras, Emmanuel Dubois	
Adaptive Hand-Tracked System for 3D Authoring	101
Alexis Heloir, Fabrizio Nunnari, Christophe Kolski	
Un espace de caractérisation pour l'édition de courbes à trois dimensions	105
Thibaut Jacob, Gilles Bailly, Eric Lecolinet, Raphaël Foulon, Etienne Corteel	
LinkWave: une Liste d'Adjacence Visuelle Interactive pour explorer les Réseaux Pondérés Dynamiques	113
Nathalie Henry Riche, Yann Riche, Nicolas Roussel, Sheelagh Carpendale, Tara Madhyastha, Thomas J. Grabowski	

Session 4 : Techniques d'Interaction : Dimensions > 2

PLACID: un planificateur pour composer dynamiquement des Services IHM	123
Yoann Gabillon, Gaëlle Calvary, Humbert Fiorino	
Un système d'aide et de suivi des tâches utilisateur dans un environnement ambiant	130
Asma Gharsellaoui, Yacine Bellik, Christophe Jacquet	
Drag-and-drop for older adults using touchscreen devices: effects of screen sizes and interaction techniques on accuracy	139
Lilian Genaro Motti, Nadine Vigouroux, Philippe Gorce	
Direct and Indirect Multi-Touch Interaction on a Wall Display	147
Jérémy Gilliot, Géry Casiez, Nicolas Roussel	

Interaction par le regard : évaluation du retour d'information progressif Van Bao Nguyen, Francis Jambon, Gaëlle Calvary	153
Un dispositif d'interaction pour favoriser la discussion et l'engagement dans les musées Patricia Plénacoste, Yvan Peter	159
Quick-Glance and In-Depth exploration of a tabletop map for visually impaired people Sandra Bardot, Anke Brock, Marcos Serrano, Christophe Jouffrais	165
BCI hybride pour la palliation du handicap moteur sévère Alban Duprès, José Rouillard, François Cabestaing	171
Predictive Usability Evaluation: Aligning HCI and Software Engineering Practices Káthia Marçal de Oliveira, Lepreux Sophie, Kolski Christophe, Ahmed Seffah	177
Challenges for Usability Testing in Ubiquitous Systems Carla Ilane Moreira Bezerra, Rainara Maia Santos, Rossana Maria de Castro Andrade, Ismayle Santos, José Maria Monteiro, Káthia Marçal de Oliveira, Mourad Abed, Houcine Ezzedine	183
Interfaces cerveau-ordinateur et jeux sérieux : adaptation aux patients schizophrènes Charlotte Piau, Marie-Hélène Bekaert, François Cabestaing	189
Impact de l'utilisabilité, la motivation et l'expérience utilisateur d'une application mobile sur l'engagement citoyen Isabelle Tissier, Guillaume Gronier	195
SuperVision : contrôler spatialement les objets connectés dans une maison intelligente Sarthak Gosh, Gilles Bailly, Robin Despouys, Eric Lecolinet, Rémi Sharrock	201
Une Lampe Anthropomorphique pour la Communication des Emotions Leonardo Angelini, Maurizio Caon, Denis Lalanne, Omar Abou Khaled, Elena Mugellini	207
Index des auteurs	213

PaperComposer: Creating Interactive Paper Interfaces for Music Composition

Jérémie Garcia^{1,2}
jgarcia@ircam.fr

Theophanis Tsandilas¹
fanis@lri.fr

Carlos Agon²
agon@ircam.fr

Wendy E. Mackay¹
mackay@lri.fr

¹INRIA
Univ Paris-Sud & CRNS (LRI)
F-91405 Orsay, France

²STMS Lab: IRCAM-CNRS-UPMC
1 place Igor Stravinsky
F-75004 Paris, France

ABSTRACT

Interactive paper technologies offer new opportunities for supporting the highly individual practices of creative artists, such as contemporary music composers, who express and explore their ideas on both paper and the computer. We introduce *PaperComposer*, a graphical interface builder that allows users to create a personalized interactive paper interface that they can connect to their own computer-based musical data. We also present an API that facilitates the development of interactive paper components for *PaperComposer*. We describe one public demonstration of a novel musical interface designed for children and our collaborations with composers to create two novel interactive music interfaces that reflected their individual composition styles.

Author Keywords

Interactive paper; music composition; creativity; toolkit

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces – Graphical User Interfaces; H.5.5 [Sound and Music Computing]: Methodologies and Techniques.

INTRODUCTION

Music composition has been deeply influenced by the introduction of composition software such as *Max* and *OpenMusic*. These tools allow composers to create new sounds, develop their own musical strategies and explore symbolic notations. However, paper remains fundamental in their creative process [11, 16, 17]. Our goal is to develop interactive systems that support composers' expression and exploration on paper while taking advantage of the computational power of computer-aided composition tools.

Contemporary music composers pose a particularly challenging design problem, because they develop their own unique musical representations on paper and also sophisticated computer programs to express their musical ideas [7, 12, 16]. Any specific solution created for one composer is thus likely to be rejected by other composers.

Rather than creating unique interfaces tailored to each composer, we want to develop flexible tools that composers can easily adapt for their own musical needs.

Interactive paper technology creates new possibilities for composers, who will be able to switch between paper and the computer, ideally combining the best of both. The most representative technology for interactive paper is Anoto¹ which combines a small infrared camera embedded close to the tip of the pen, and a tiny, almost invisible, dot pattern printed on paper. The camera detects the pattern and identifies the precise location of the pen on the page while the user is writing.

We are interested in developing a simple toolkit that supports the creation of diverse music-based interactive paper applications. Our goal is to handle different forms of musical data, e.g., musical scores, graphs and personalized musical representations, and link them to a variety of music composition applications, incorporating the standard communication protocols and data structures used by professional composers.

Our previous research introduced the concept of *Paper Substrates* [8]: paper-based composition elements that manage musical data according to specific rules. Composers can create their own personal musical structures and link them to musical data types on the computer. The composer can interact with these musical structures via interactive paper, which can accept and update data from computer composition tools. The first version of *paper substrates* offered a limited set of interactive components and required significant programming to create new ones, limiting the ability of end users to tailor the interface. The next step is to create tools that facilitate the creation and deployment of new *paper substrates*, so that composers can adapt them for their own musical needs.

We introduce *PaperComposer*, an interactive interface builder for creating and customizing *paper substrates*. We describe the API that supports the creation of paper components and present specific examples of interactive paper composition applications created with *PaperComposer*. We discuss the current implementation and conclude with suggested directions for future research.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670450>

¹ <http://www.anoto.com>

RELATED WORK

Researchers have developed a variety of interactive paper solutions for diverse domains. For example, Paper-Link [2] lets users create hyperlinks from physical documents to electronic content. Strip'TIC [14] is designed to support air traffic controllers by augmenting paper flight strips with pen-based and tangible interactions, and includes visual feedback overlaid on the strips.

Musically oriented interactive paper applications pose special challenges, since composers often sketch on paper to express personal or ill-formed musical ideas that cannot be represented easily with traditional musical representations [11, 12, 16, 21]. For example, Musink [22] helps composers define their own annotations with respect to musical scores and link them to computer-based functions. InkSplorer [7] lets composers draw hand-written curves to control computer programs and PaperTonnetz [3] lets composers explore melodies and chords within hexagonal-shaped sets of tone networks. *Paper substrates* [8] support pen input and other forms of tangible interaction with the paper interface, such as moving, associating and layering paper components to create higher-level musical entities. *Paper substrates* handle musical data from computer-aided composition applications and provide visual structures to support data entry, data editing, and interaction with the pen.

Creating advanced interactive paper applications from scratch requires significant programming effort. Commercially available software development kits offer basic functionality for creating interactive paper applications with Anoto technology, but focus primarily on document management, especially note-taking, editing printed documents and filling out forms. Anoto offers an Adobe Acrobat plugin that allows end users to create, print and interact with forms that content text fields and check boxes. Livescribe offers somewhat more functionality via an Eclipse plugin that lets developers place arbitrary shapes onto a paper interface and link them via pen events. However, this tool requires additional Java code to interpret pen events and create standalone pen applications, and the user must install both the pen application and the paper interface on the pen in order to use them.

Researchers have explored several frameworks and toolkits that speed development and support creation of more complex interactive paper applications. For example, iPaper [18] is a framework that supports rapid development and deployment of interactive paper applications. Programmers define active areas or specialized components and link them to specific services such as sending an email with handwritten content. Paper Toolkit [24] is a generic event-based architecture and toolkit for developing paper user interfaces. Letras [13] assists the deployment of paper applications where interaction is distributed across different input and output devices. Each of these programming environments is designed to help experience programmers to develop highly specialized applications. Unfortunately, they are not suitable for

composers or other end users, because they require general programming skills. Note, however, that composers *do* learn to use certain specialized music composition programs, such as *Max* and *OpenMusic* [1]. These allow users to visualize and assemble higher-level musical primitives into networks of numerical values, controllers, graphs, as well as handle classical music notation. Composers also use the Open Sound Control (OSC) protocol [23] to exchange musical data across applications.

Paper Composer is designed for composers who are familiar with these standard, higher-level music composition programs. They can create personalized interactive paper interfaces that control these higher-level musical primitives, which enables them to explore ideas informally on paper and then fluidly move to controlling them on the computer. The goal is to extend existing computer-based composition tools to support the early and intermediate phases of creative exploration, with smooth transitions between simple marks on paper to complex interactive functions on the computer.

PAPERCOMPOSER: AN INTERFACE BUILDER FOR INTERACTIVE PAPER

Our goal is to enable composers, as end users, to create, manage and use their own interactive paper interfaces without necessarily having to program. We decided to build upon the concept of *paper substrates* [8] since they allow end users to create personal structures on paper for specialized musical content which is then interpretable by music-programming software. Previous studies [7, 8, 16] show that composers appreciate the new possibilities offered by interactive paper but also require support for their current composition environments. They also require functionality that simplifies stroke recognition and supports archiving and reuse.

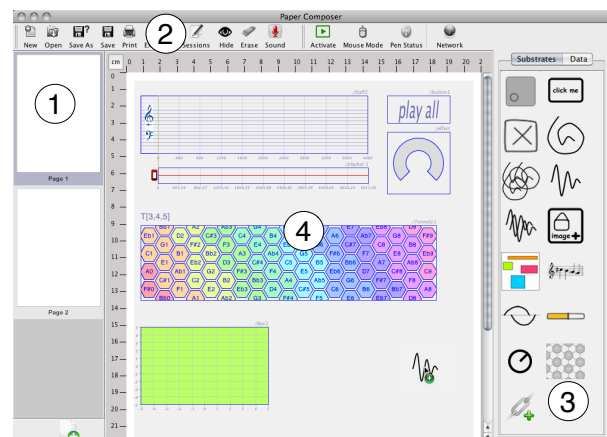


Figure 1: PaperComposer interface

- (1) Current document pages.
- (2) Toolbar to manage document and set parameters.
- (3) Thumbnails of available component classes can be dragged into the virtual page to create new instances.
- (4) Virtual page with components: musical staves, play-bar, button, knob slider, tonnetz, curve container.

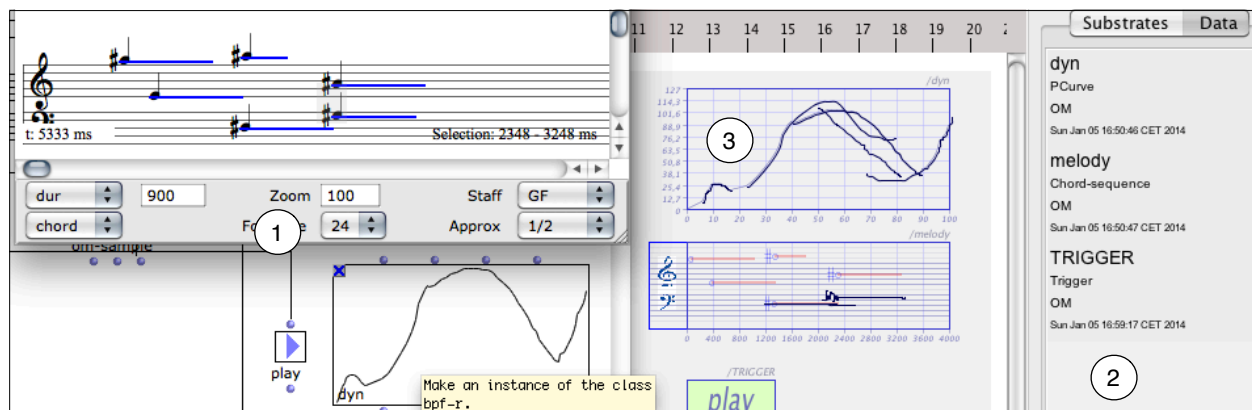


Figure 2: Synchronizing data between an external music tool (OpenMusic) and PaperComposer.
 (1) OpenMusic patch with a graph and chord sequence, exportable via a custom OpenMusic library.
 (2) List of imported data from OpenMusic that can be dragged to any component.
 (3) Graphical representations of the components, data and strokes drawn on paper.

To achieve these goals, we developed *PaperComposer*, an interface builder for creating paper components that represent *paper substrates* and control digital musical classes. Users can create specialized components with various properties and connect them via OSC data channels to external applications. They can combine multiple components into custom page layouts and test the interaction with a mouse before printing the page on paper with the Anoto dot pattern. *PaperComposer* provides live feedback as the user uses an Anoto pen to draw on the page, and enables the user to store and reuse previously drawn pen strokes.

Document Creation and Customization

Figure 1 illustrates the creation of a two-page document in portrait format. The user can add components by dragging them from a list at the right side of the page window. The user can resize and position a component on any page and then parameterize it through a configuration window activated with a right click. Parameters that can be reconfigured include the color of the component and its output OSC channels as well as component-specific parameters, such as a range of values and duration, if the component represents a musical sequence. The user can define static links between components represented by straight non-printable lines. Components can accept data from compatible data sources, both files and external applications.

Figure 2 shows how data exported from OpenMusic appears in a list in *PaperComposer*. In this scenario, the user decides to edit an existing musical sequence in OpenMusic by adding new notes, modifying the duration of existing notes and controlling their dynamics with an amplitude curve. The user exports three different data streams from OpenMusic: a curve, i.e., a list of points, a musical sequence, and a trigger to consume events such as a pen down to launch the play command. The user can establish an association by dragging any data source element from the list and then dropping it onto a compatible component. If the user drops the data onto a blank

part of the page, a menu appears with all compatible components. The user makes a choice, which creates a new paper substrate that hosts the associated data.

The *PaperComposer* components that appear on the screen can be printed directly onto interactive paper with an automatically superimposed Anoto dot pattern. *PaperComposer* also supports the creation of interactive paper interfaces from existing scores created with musical editors such as *Finale* and *Sibelius*. Each page of the score becomes a new document page with associated interactive regions for each score element. For example, each note is interactive. When the pen touches a note, *PaperComposer* plays the corresponding pitch.

Deployment, Debugging and Session Management

PaperComposer displays the users interactions on paper in real time on the screen, including the current page and the strokes being drawn on paper. Figure 2 shows the paper interface at runtime, overlaid with the current strokes as well as how the associated musical objects were updated in OpenMusic.

All strokes and audio recordings captured during a session are automatically stored and labeled for later reuse. The user can use the session manager from the menu bar to manage past sessions, load them onto the screen and use components to reinterpret them. Developers can also test newly created components with data of older sessions and iteratively improve their components by reusing recorded strokes as input. Finally, *PaperComposer* supports the monitoring of the network activity between the paper application and musical software. The user can open a monitor window by clicking on the network icon of the top toolbar and track connected client applications and OSC messages exchanged through the network.

Examples of Interactive Components

Figure 3 gives an overview of paper components currently available in *PaperComposer*.

1. *Knob slider*: controls continuous values in real time.

2. *Button*: triggers actions in response to one or more pen events: down, up, moved, entered, exited, tap, and double tap.
3. *Curve container*: supports entering curves and provides a customizable grid layout to reinforce different levels of precision. One type of curve container supports entering a single curve such that each new stroke is merged to the curve, incrementally changing its trace. The other type supports creating collections of curves, such that each new stroke represents a different curve.
4. *Musical sequence container*: supports interaction and editing of timeline representations of musical sequences [8].
5. *Playbar*: selects time ranges to play specific parts of a musical sequence.
6. *Tone network*: supports the interactive creation of musical sequences of both chords and melodies [3].

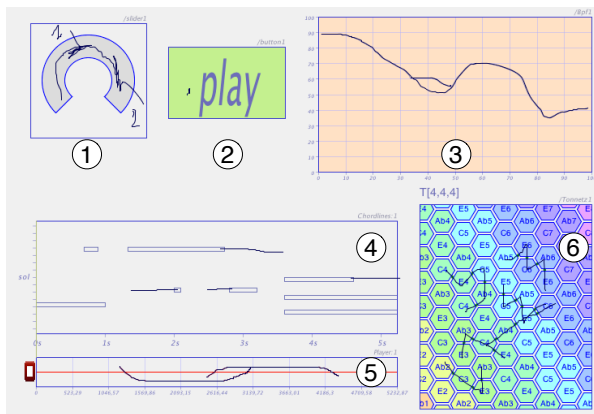


Figure 3: Examples of existing components.

AN API FOR PROGRAMMING PAPER COMPONENTS

We created an application-programming interface (API) that provides classes for extending *PaperComposer* with new data representations and interactive components. It includes classes for creating and printing customized templates, processing typed data, storing data and sessions, handling pen events and pen gestures, and receiving or sending messages to external applications. Figure 4 summarizes the architecture of the API.

Main API Components

The API allows the developer to easily create paper applications by defining instances of existing classes of paper components on paper coordinates of one or more pages. The developer can also create new types of components by extending the *PComponent* class. The API takes care of the management of the Anoto dot pattern, the management of sessions, and the communication of the components with external applications. We use *Java*

Reflection to load new component definitions and their instances at run time. Multiple components can be grouped together into a single paper application represented by the Document class. A document specifies a set of pages, where each page can contain several components. We have developed utilities to store, edit and load a single or multiple documents for live interaction. All pen gestures created by users at runtime are stored within sessions for later reuse and archiving purposes. In addition, the API supports recording audio synchronized with the captured pen data.

A component is a composite paper widget, similar to the first-class widgets supported by common user interface toolkits such as Java Swing. It has a shape and position on paper expressed in millimeters and has a visual representation on screen and on paper. More specifically, it provides methods responsible for the painting and printing of its graphical elements and data. These methods are compliant with *Java2D*, letting developers reuse existing *Java2D* code. A component can be associated with coordinate transforms that map its data from millimeters to application-specific values. It can also register to listeners of specialized pen events such as pen down, pen entered, stroke created, or gesture recognized that happen within its shape.

A component can contain one or more interactive components, where each component can have its own listeners. Printed and handwritten data can also be added as child components that listen to pen events. Since printed pages may contain large numbers of components that respond to pen events, we developed an event manager to store and retrieve components and data by using efficient spatio-temporal indexing techniques. We use the R-Trees implementation by the SaIL Library [10].

Communication among Paper Substrates

We designed communication mechanisms that let us create complex musical data from several layered or connected components. Components can communicate with each other through static and dynamic links [8]. Static links are created either programmatically or by using the graphical interface of *PaperComposer*. Dynamic links are created interactively with the pen by drawing a stroke that crosses two different components.

Users can only create links among compatible components. One of the linked components usually acts as a master that defines the reference of data sharing between them. The link mechanism also enables a paper component to reuse strokes written on other components, re-interpreting, rescaling and transforming them before re-directing them to specialized patches on the computer.

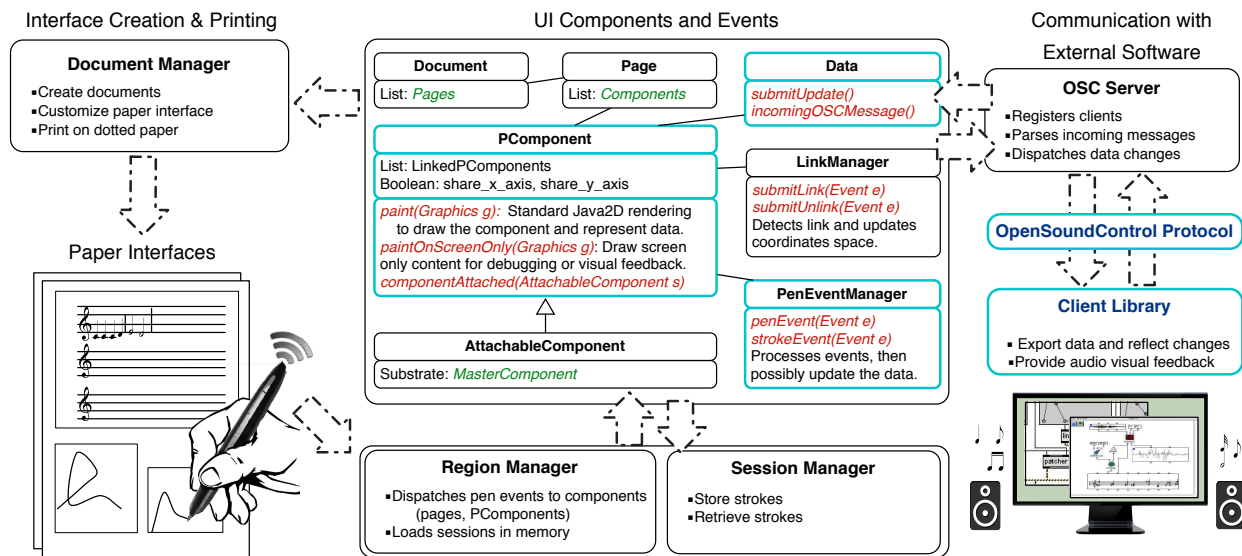


Figure 4: PaperComposer programming API supports creation of paper interfaces to manipulate musical data. Developers edit blue elements to identify data types, specify data representations and define user interactions.

Communication with External Music-Programming Software

Components can exchange musical data with external applications using the OSC protocol [23]. An OSC server allows registered applications to send and receive updates from components. We define an OSC-based data exchange protocol for a variety of data types, from basic lists of numerical values to more complex sequences music notes and chords. Data types are extensible and developers can add new data types or implement new components to handle them. PaperComposer can connect components with Max and OpenMusic libraries, chosen because they widely used by composers and can be easily extended programmatically. The OpenMusic library is implemented in Common Lisp and supports common musical objects such as graphs for controlling musical parameters and basic score containers. These objects can export their data to interested components. The Max library is written in JavaScript as an external object and can be used within any Max patch to export its window and interface elements, e.g., buttons, graphs and sliders. Developers and users can extend the two libraries to add new data types and functionality.

Support for Musical Scores

The PaperComposer API also supports the use of scores from commonly used music notation editors such as Finale and Sibelius. A dedicated class loads the formal description in the MusicXML² format with the PDF of the score. Each graphical element of the score is associated with an interactive area on the paper page and related to the corresponding musical content, using the ProxyMusic Library³. We also created a class to play the correspond-

ing MIDI note if the pen enters a particular note’s area. Components can also use the data from the score to create alternative representations and export the result into a new MusicXML file that can be processed and rendered by most score editors.

Component Creation Process

Developers follow these steps to implement and deploy a new paper substrate or build their own interface:

Step 1: Create a class that inherits the PComponent class or the AttachableComponent class.

- Set an icon file and a name.
- Set the coordinate space and set Booleans to indicate how axes should be synchronized when links with other components occur. This allows a component to use the axis and scale of a linked master component and apply it to its data.
- Set the accepted data types.
- Program the paint method with standard Java2D instructions.
- Add one or several pen event listeners and implement how the events modify the data.

Step 2: Create a class that inherits the Data class.

- Define an OSC communication protocol.
- Implement incomingOSCMessages() & submitUpdate() methods.

Compatible data sent by an external application can be loaded into any component. Touching a component with the pen triggers a data notification to the application.

We provide several utilities to simplify stroke recognition, including using a distance threshold, Douglas and Peucker’s algorithms [5], and extracting geometrical features, including main orientation, duration, intersections, from other strokes. Interpreting pen input remains challenging due to the high variability in how users write

² <http://www.musicxml.com/>

³ <https://proxymusic.kenai.com/>

with respect to temporal order and shape. *PaperComposer* supports the use of *Java* libraries and available toolkits such as *iGesture* [19] to recognize gestures.

EXAMPLES OF PAPERCOMPOSER APPLICATIONS

We used the *PaperComposer* API to design and implement several interactive paper applications.

Tangible Scores for Electro-Acoustic Composition

We explored how physically positioning interactive pieces of paper can be used to create musical sequences. We developed a tangible score application that lets users create musical sequences by assembling several sounds on a paper interface.

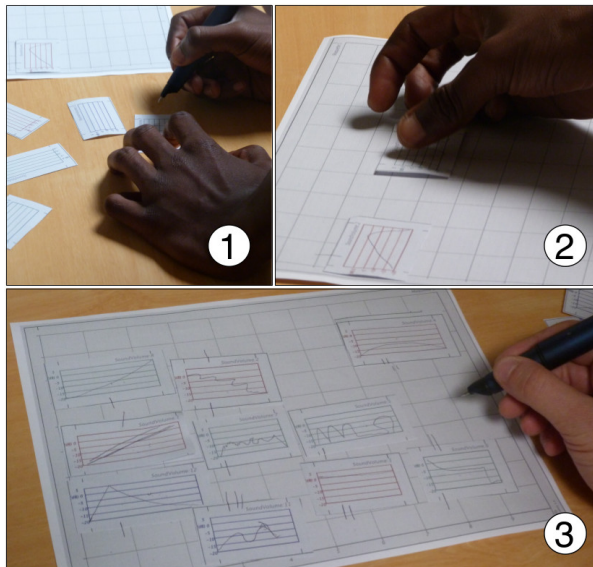


Figure 5: Creating a score with movable substrates.

- (1) Draw a curve on a sound substrate to control amplitude.
- (2) Position a sound substrate to define onset time.
- (3) Touch the score substrate to listen to the resulting score.

Interactive paper design: The application includes a paper substrate that takes a whole page and represents the space of the score where other small substrates linked to sound files can be added. Each small substrate has a color that identifies a sound and a width that defines its duration. The user can draw curves control the amplitude of the sound over time. The user can also assemble the small elements into the main sequence by sticking them onto its 2D space and creating a dynamic link between the two substrates with the pen. This creates a collage of sounds. The user can tap the pen on any individual paper substrate to play the associated sound or tap on the score to hear the complete piece. The paper interface communicates with the *Max* patch responsible for amplifying and mixing the sounds based on instructions given by the paper substrates. This application is a simplified but tangible representation of *OpenMusic*' [4] maquette interface, which provides a spatial layout to help musicians mix programmable musical objects together and organize them over a timeline.

Public demonstration: We demonstrated the user interface during the *Fête de la Science 2011* (see Figure 5). We encouraged participants to explore the properties of different sounds and understand the goals and tools of computer-assisted music analysis and composition.

Approximately 60 to 70 participants, mostly children, interacted with the interface to create collaborative musical pieces. They enjoyed editing their scores with the pen. Several explored alternative results by spatially rearranging their sound components and by drawing different curves on *paper substrates* of the same color. We observed that the tangible nature of the interface is particularly useful for making music creation tools accessible to children in a playful manner.

Proportional Musical Notation

We collaborated with a graduate student in composition at Ircam who had created a personal musical notation for a composition for alto and electronics. Using a participatory design approach, we used *PaperComposer* to iteratively design a *paper substrate* that recognizes handwritten notes over a staff, lets him to listen to the musical result and transfer it to *OpenMusic* or *Max*.

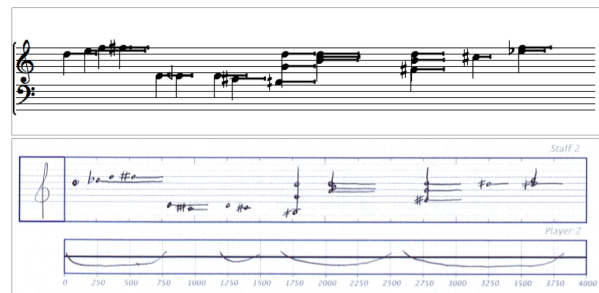


Figure 6: Paper substrate co-designed with a composer.

- Top: Resulting musical score in a *Max* patch.
- Bottom: Handwritten notes represent pitch and duration. Playbar substrate selects, evaluates individual score segments.

Figure 6 presents the score written by with his personal notation and the resulting musical sequence in *Max*. According to this notation, notes have an absolute position in time, which is defined by their horizontal position within the staff. A horizontal line nested to a note represents its duration. Our recognizer is based on simple heuristics and supports the following elements: keys (G, F and C alto), notes, intonation symbols (quarter tones, sharps and flats) and duration lines. The composer's original notation does not include stems, i.e., vertical bars that group notes into chords. We added stems to simplify the identification of chords and their onsets. The user can evaluate the musical result by tapping on the rectangular area around the key symbol or by using an additional playbar substrate. To correct a recognition error, the user can hold the pen over the misrecognized symbol and redraw it.

The composer created his own interface components with *PaperComposer*. He combined the notation substrate with several graph substrates to control the dynam-

ics of his musical sequences. He successfully created *Max* patches that react to the data transmitted from the pen to produce a final sequence played by his own synthesizer.

Recognizing handwritten musical notes is a challenging problem [15]. Previous work has tried to tackle it by using simplified gesture vocabularies [6] or interactive devices that allow the user to supervise the recognition process and correct errors [21]. The present *paper substrates* considerably simplify the recognition problem by using a single type of note with no special notation for rhythms or rests.

Controlling Sound Synthesis with Pen Gestures

We collaborated with another composer to design a *paper substrate* for drawing gestures that control a sound synthesis engine developed by the composer in *Max*. According to his composition framework, the visual space acts as a sequencer where the horizontal axis represents time. The composer organizes several musical elements in time by positioning them in the patch space. He mainly uses text-based descriptions of synthesis parameters but is interested in enriching his expression possibilities with pen input.

We worked with the composer to create a *paper substrate* that maps dynamic gestures drawn on paper to the dimensions of computerized objects in a *Max* patch. The actual properties that form the profile of a gesture are the list of its point coordinates with their timestamps and the associated pen pressure values. We use color-coding to reveal the orientation of a gesture, and visual information that the composer uses in the synthesis engine. The beginning of each stroke appears as blue and progressively changes to red.

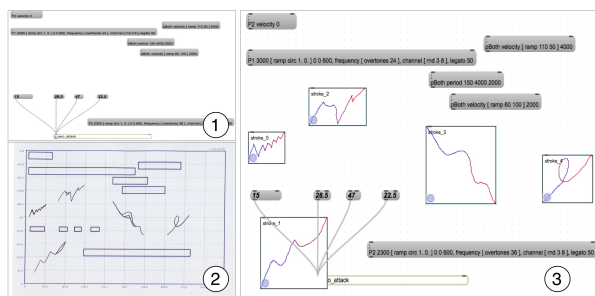


Figure 7: Controlling sound synthesis with pen gestures.

- (1) Original score in a *Max* patch.
- (2) Score substrate with handwritten gesture refinements.
- (3) Resulting score in a *Max* patch.

The *paper substrate* represents existing score objects as rectangles that guide the drawing of the synthesis gestures. After expressing a gesture on paper, the composer can refine it by redrawing only parts of the gesture. In this case, the refining stroke is merged with the initial one. We carefully update the timestamps of the new data points so that they preserve the dynamics of the original gesture. This allows the composer to explore variations of a gesture's shape without changing its original dynam-

ics. He can also edit the position of a gesture in the electronic view and then reprint an updated version of the *paper substrate* in order to continue his work on paper.

Figure 7 illustrates how the composer uses the *paper substrate* as part of his workflow. First, he exports sequences of musical objects from his *Max* patches. He then uses *PaperComposer* to associate the exported data streams with a specialized sequence substrate. He prints the interface on paper to draw his synthesis gestures. The *Max* patch shows the new gestures immediately after they are drawn on paper.

DISCUSSION AND FUTURE DIRECTIONS

PaperComposer supports the implementation and deployment of powerful user interfaces for working with music on paper. We observed that linking music programming software with paper applications could make such environments more accessible to children and novice users. At the same time, our approach can help professional composers reflect on the implementation of their ideas on paper while interacting with their composition tools.

PaperComposer requires the user to define the whole interface on the screen before printing and using it. However, studies have shown that composers frequently use paper interfaces for ill-formed ideas and iteratively refine both their ideas and their notation [7, 12, 16]. Therefore, we also need to consider the interactive definition of *paper substrates* by drawing them directly on paper before or after writing musical content. For example, the Livescribe pen allows users to draw a piano keyboard made of vertical and horizontal lines anywhere on the page. Once the piano is drawn, the pen can play the corresponding note when the pen tip enters a key. We are interested in extending our tools to offer similar functionality by enabling users to define interactive structures, such as an interactive musical staff, by drawing directly on paper. This may help composers better integrate paper interfaces in their composition process with incremental transitions between ill-formed ideas and more formal ones.

The *PaperComposer* toolkit provides assistance for programming new components and for creating and customizing end-user interfaces. Composers can build personal interactive-paper interfaces for a range of music-creation tasks: drawing and refining sound-synthesis curves, writing musical sequences by using music notations or Tonnetz representations, controlling music software with paper buttons, timeline selectors, playbars and sliders.

To extend the collection of paper substrates supported by *PaperComposer*, we met with composers and examined the representations that they use to write music. We collaborated with them to design new *paper substrates* that assist or enhance specific aspects of their work process. We implemented these *paper substrates* with our toolkit. Composers could then use them within *PaperComposer* to connect their paper-based activities with their comput-

er-aided composition environments. More recently, we used the toolkit to create *Polyphony*, which was tested on a short composition task by 12 music composers [9].

Currently, we support only audio and visual feedback via external applications in response to events and data transmitted by the pen. However, we are also interested in supporting additional feedback modalities and richer forms of physical interaction with pen and paper, such as bimanual interaction techniques based on the use of portable mini-projectors [20] and mobile phones [21]. Such approaches can bring visual feedback closer to the space of writing. We are also interested in extending the API and *PaperComposer* to enable programmers and users to define or reconfigure such forms of audio and visual feedback at the level of the actual paper components. This will result in richer paper applications that mix both the physical and the virtual instances of a paper component. A longer-term goal is to explore the dynamic definition of new interactive substrates, either on paper or on the screen, with visual programming languages, since many composers are familiar with such languages.

SUMMARY AND CONCLUSION

We developed a toolkit that includes a user interface builder, *PaperComposer*, and an API. *PaperComposer* allows both developers and end-users to create interactive paper interfaces with specialized paper components, connect the components with data streams of external musical software, customize their appearance and deploy their interactive paper applications. It also provides mechanisms for session management, debugging, and logging. The API enables developers to define new paper components that accept additional musical data with their own representation structures and interactions.

We used these tools to create new paper interfaces for musical creation. We developed and demonstrated a paper application for creating electro-acoustic scores based on the spatial positioning of movable paper components. We also collaborated with composers to design interfaces for writing music notation and controlling sound synthesis. These examples demonstrate the practical use of our toolkit. Future work needs to further evaluate it through workshops with developers of music applications. We are also interested in using it in longer field studies with professional composers and musical assistants.

ACKNOWLEDGMENTS

We are grateful to all the composers who helped us improve *PaperComposer*. We would also like to thank the researchers from the Musical Representations team at IRCAM and the InSitu team at LRI for their help and insights. This work was supported in part by ANR project INEDIT (ANR-12-CORD-0009).

REFERENCES

- [1] Agon, C., Bresson, J. and Assayag, G. 2006. *The OM composers' book Vol.1 & Vol.2*. Collection Musique/Sciences.
- [2] Arai, T., Aust, D. and Hudson, S.E. 1997. PaperLink: a technique for hyperlinking from real paper to electronic content. In *Proc. ACM/CHI '97*, 327–334.
- [3] Bigo, L., Garcia, J., Spicher, A. and Mackay, W.E. 2012. PaperTonnetz: Music Composition with Interactive Paper Tonnetz. In *Proc. SMC'12*, 219–225.
- [4] Bresson, J. and Agon, C. 2008. Scores, Programs, and Time Representation : The Sheet Object in OpenMusic. *Computer Music Journal*. 32-4, 31–47.
- [5] Douglas, D.H. and Peucker, T.K. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*. 10, 2, 112–122.
- [6] Forsberg, A., Dieterich, M. and Zeleznik, R. 1998. The music notepad. In *Proc. UIST '98*, 203–210.
- [7] Garcia, J., Tsandilas, T., Agon, C. and Mackay, W. 2011. InkSplorer: Exploring Musical Ideas on Paper and Computer. In *Proc. NIME '11*, 361–366.
- [8] Garcia, J., Tsandilas, T., Agon, C. and Mackay, W. 2012. Interactive paper substrates to support musical creation. In *Proc. ACM/CHI '12*, 1825–1828.
- [9] Garcia, J., Tsandilas, T., Agon, C. and Mackay, W.E. 2014. Structured observation with polyphony: a multifaceted tool for studying music composition. In *Proc. ACM/DIS '14*, 199–208.
- [10] Hadjieleftheriou, M., Hoel, E. and Tsostras, V.J. 2005. SaLL: A Spatial Index Library for Efficient Application Integration. *Geoinformatica*. 9, 4, 367–389.
- [11] Hall, P. and Sallis, F. 2004. *A handbook to twentieth-century musical sketches*. Cambridge University Press.
- [12] Healey, P.G.T., Jean-Baptiste Thiebaut and Thiebaut, J.-B. 2007. Sketching Musical Compositions. In *Proc. CogSci '07*, 1079–1084.
- [13] Heinrichs, F., Steimle, J., Schreiber, D. and Mühlhäuser, M. 2010. Letras: an architecture and framework for ubiquitous pen-and-paper interaction. In *Proc. ACM/EICS '10*, 193–198.
- [14] Hurter, C., Lesbordes, R., Letondal, C., Vinot, J.-L. and Conversy, S. 2012. Strip'TIC: Exploring Automatic Paper Strip for Air Traffic Controllers. In *Proc of AVI '12*, 225.
- [15] Lee, K.C., Phon-Amnuaisuk, S. and Ting, C.Y. 2010. A comparison of HMM, Naïve Bayesian, and Markov model in exploiting knowledge content in digital ink: A case study on handwritten music notation recognition. *IEEE International Conference on Multimedia and Expo*, 292–297.
- [16] Letondal, C., Mackay, W.E. and Donin, N. 2007. Paperoles et musique. In *Proc. ACM/IHM*, 167–174.
- [17] Newman, T.U. 2008. *The Creative Process of Music Composition: A Qualitative Self-study*. ProQuest.
- [18] Norrie, M.C., Signer, B. and Weibel, N. 2006. General framework for the rapid development of interactive paper applications. In *Proc. CoPADD*. 6, 9–12.
- [19] Signer, B., Kurmann, U. and Norrie, M. 2007. iGesture: a general gesture recognition framework. In *Proc. ICDAR '07*, 954–958.
- [20] Song, H., Guimbretiere, F., Grossman, T. and Fitzmaurice, G. 2010. MouseLight: bimanual interactions on digital paper using a pen and a spatially-aware mobile projector. In *Proc. ACM/CHI '10*, 2451
- [21] Tsandilas, T. 2012. Interpreting strokes on paper with a mobile assistant. In *Proc. UIST '12*, 299.
- [22] Tsandilas, T., Letondal, C. and Mackay, W.E. 2009. Musink: composing music through augmented drawing. In *Proc. ACM/CHI '09*, 819–828.
- [23] Wright, M. and Freed, A. 1997. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In *Proc. ICMC'97*, 101–104.
- [24] Yeh, R.B., Paepcke, A. and Klemmer, S.R. 2008. Iterative design and evaluation of an event architecture for pen-and-paper interfaces. In *Proc. UIST '08*, 111.

A design space of guidance techniques for large and dense physical environments

Hind Gacem^{1,2}Gilles Bailly^{2,1}James Eagan^{1,2}Eric Lecolinet^{1,2}¹ Télécom ParisTech, ² CNRS LTCI UMR 5141, Paris, France

{firstname}. {lastname}@telecom-paristech.fr

ABSTRACT

Finding an object in a physical environment is difficult if the environment contains many objects, especially if it is large and dense. We propose a design space that describes and compares existing guidance techniques according to four dimensions: output modality, physicality, granularity and spatial information. Output modality can be visual, audio or tactile. Guidance information can be displayed using physical objects or virtual artifacts. Granularity indicates whether the technique serves to navigate towards the vicinity of the target or to precisely localize the target. Finally, spatial information is either exocentric or egocentric. This design space aims at providing an overview of the domain and helping designers and researchers to understand the key properties of these techniques. It also enables their comparison and the generation of new techniques by highlighting unexplored areas.

Mots Clés

Visual search; Navigation; Guidance Techniques; Design space;

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Locating an object in a physical place is a common activity in the real world. Such a task can quickly become difficult if the environment contains many objects, especially if it is large

and dense. For instance, libraries, supermarkets, factory plants, etc., can contain thousands of objects with many of them very close to each other. Moreover the size of these environments makes the problem harder [1] as the user must not only find the desired object in the proper area, but also find where this area is located in the environment.

Various guidance techniques have been proposed in the literature (e.g. [24, 13, 48]) for finding objects in such environments. However, the literature does not systematically describe (1) the differences and similarities of these techniques, (2) their relative advantages and drawbacks, and (3) their interactions with the different output modalities and the complexity of the environment.

In this paper, we propose a design space to describe and compare existing guidance techniques for indoor search tasks in a large and dense environment. This design space is organized according to four dimensions. The first dimension corresponds to the *output modality* used by the technique. In addition to the traditional visual, audio and tactile categories, we also attempt to take into account the impact of *physicality* through a second dimension, which corresponds to the case when guiding information is embedded into physical objects.

The third dimension of our design space is the *granularity* of the technique, which can either provide *micro* or *macro* guidance. This axis is important because it corresponds to two different kinds of tasks and user behaviors and thus to different classes of techniques that do not require the same level of precision and

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670455>

resolution. Moreover, this dimension also helps understand how these techniques can be combined to build hybrid techniques that provide both macro and micro guidance. Finally, the fourth dimension indicates the type of spatial information that is provided by the technique, which can be either exocentric or egocentric.

Previous explorations of such guidance techniques often treated these axes independently, especially for what concerns granularity. Our key insight is that many search tasks involve both the macro and micro phases and should thus provide hybrid guidance techniques that combine different modalities and spatial relationships depending on the local granularity.

In this paper we present a design space that is capable of reflecting these hybrid guidance techniques. More specifically, the aims of this paper are 1) to provide an overview of the field of guidance techniques; 2) to help designers and researchers understand the key properties of these techniques, compare them and choose the one that fits their needs; 3) to invite researchers to investigate unexplored areas and to generate new, possibly hybrid, techniques.

DESIGN SPACE

Guidance techniques help the user to find, locate, and reach a target without having to perform an exhaustive search within the space. They generally have two characteristics: 1) they orient the user's body movement, perhaps iteratively, to reach the target location, and 2) they indicate the target location.

These orientations and indications may use various *modalities* to present information, such as visually, using sound, etc. They may be embedded in physical objects or use virtual hints. They may operate at different *granularities* depending on whether the target in the user's field of view or on the size of the target. Finally, the technique may provide these cues using exocentric or egocentric *spatial relationships*. Together, these four axes make up the dimensions of our proposed design space, which we describe below.



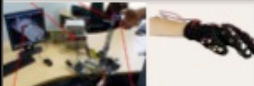



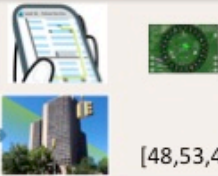



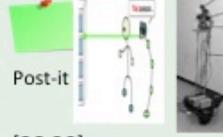
We further provide two representations of this space, depicted in Figures 1 and 2. The first representation is modality-centric and is especially useful for identifying comparable guidance techniques. The second is granularity-oriented and is better suited to the hybrid guidance techniques that combine different techniques for different granularities.

In the remainder of this section, we introduce these dimensions and related work. In the next sections, we describe these dimensions in more detail, and explore two representations of this design space.

Modality. The modality is the sensory modality [35]. It is used to convey its guidance hints to the user. Of the five human senses (sight, sound, touch, smell, and taste), we didn't include the last two, in the design space representation, because techniques based on olfactory and gustatory interaction are still in an early stage of development [27] and hence seldom used in augmented reality systems [55].

Physicality. Beyond these base modalities, physical objects have *presence*, which may be perceived via any of these senses or a combination of them. Physical objects provide these modalities in a consistent manner and with a higher level of realism than virtual artifacts. For instance, Danieau showed that combining virtual modalities (audio, visual and tactile) enhances perception but still does not provide the same feeling of realism [7]. Moreover, physical objects do not require users to wear AR devices (headphones, head-mounted displays, force-feedback exoskeletons, etc.) that may be intrusive or cumbersome and may limit users.

Granularity. Indoor search tasks consist of two primary subtasks: macro navigation and micro navigation [22,50]. Macro navigation generally corresponds to the coarse phase of locating the vicinity of the target, such as the shelf on which a book resides. Micro navigation involves the finer-grained location of an object already within the field of view, such as finding a jar of jam when staring at an open refrigerator.

Modalities	Tactile	 Left/right vibration on wrist (waist) [21,18]	 NEWS directional tap ,... [54]	 Tactile glove – surgery tool with force and thermal feedback [14]	 [32]
	Auditory	Panned (frequency, amplitude,...) Speech (realtime positioning) [52,57]	3D sound, speech,... [12,13,15]	 3D sound,... [45, 26]	1D sound Bip left/right,... [39]
	Visual	 Left/right Arrow [34,4,16]	 [48,53,4]	 [33,16,41,36,5]	 Left/right Arrow, Text[40]
	Physical		 Arm pointing	 Post-it [38,29]	
	Egocentric	Exocentric	Exocentric	Egocentric	
	Macro		Micro		

Type of guidance technique (Macro/Micro, Exocentric/Egocentric)

Figure 1: First representation of the design space, which classifies guidance techniques according to modality, physicality, granularity (macro/micro) and the spatial guidance type (egocentric/exocentric). Physicality and modalities are two orthogonal dimensions represented on the same axis to simplify the visualization.

Spatial Guidance Type. The spatial guidance type describes the frame of reference in which the system provides guidance, which may be *exocentric* or *egocentric*. Exocentric guidance provides information related directly to a position or a direction in the environment, be it the user, the environment, or the target. Egocentric guidance, on the other hand, provides only a partial guidance, relative to the position and orientation of the user, such as by a vibration on a wrist strap indicating to turn right [21]. In this latter case, the guiding information is generally provided iteratively to tell the user in which direction he should go or look for the target.

Related Work

Many studies propose guidance techniques to localize virtual targets in virtual environments

(e.g. [37,8]), we focus on the guidance and localization of real objects in the real world.

Several studies have attempted to characterize various guidance techniques. We are not aware, however, of other design spaces of guidance techniques. Buchmann et al. [4] have characterized guidance techniques according to the type of *information*. They distinguish between angular (compass) and non-angular information (left/right arrow) and between on-body (direct) and off-body (indirect, on external devices) techniques. The first of these types of information fits well within our definition of the spatial guidance type. Furthermore, Buchmann et al. focus on macro guidance techniques and do not take into account micro ones. For instance, their classification does not adequately account for

the Shelftorchlight [36], which uses micro navigation guidance.

Guidance techniques typically rely heavily on indoor localization, for which many characterizations exist (see [47,49] for a literature review). However, these works focus more on algorithmic and technological aspects rather than on the higher-level interaction and guidance.

We now present several guidance techniques to illustrate our design space. They are organized according to our four axes.

DESIGN SPACE DESCRIPTION

In constructing this design space, we aimed to take a person-centric approach. The user and his or her goal of locating an object guide the design of the four dimensions: the user's senses inform the modality; the way of displaying information informs the physicality; the relationship between the user and the target informs the granularity; and the frame of reference between the two informs the spatial guidance type.

We purposely avoid directly considering technology among these axes. Different tracking systems have different characteristics of precision, accuracy, and scale, and thus limit which guidance techniques can be used with which techniques. Different presentation technologies such as head-mounted displays, projectors, vibro-tactile wristbands, etc., further influence which modalities, granularities or spatial guidance types are appropriate.

Together, these four dimensions serve to provide an abstract overview of augmented reality guidance techniques. In the remainder of this section, we consider these dimensions and influences in more detail.

Modality

Each of the five human senses offers different properties that make it more or less suitable for different kinds of tasks [13]. For example, vision offers a higher information bandwidth than audio [20], but audio can remain effective when a target is beyond the field of view.

Visual. Visual guidance techniques are perhaps the most common of the modalities. They may use text [40,51] light [36] or shape [33] in 2D or 3D space [16]. Furthermore, displayed information may be static or animated.

Audio. Auditory guidance techniques map sound parameters [17] such as frequency [57], amplitude and rhythm [2], or use auditory icons [11,17], earcons [42,17], vowel sound mapping [15], spatialization through 3D sound [30,52], or combinations of sonification and spatialization [37,12] to convey guidance information. They may alternatively use speech synthesis to provide verbal guidance information. Using 3D spatial mapping techniques, in headphones [30] or in speaker networks [45], can provide a rich, natural spatial mapping [12]. Alternatively, simple stereo headphones can provide for more basic left/right guidance relative to the user.

Tactile. Tactile guidance techniques use the tactile feedback perceived by the human body. Tactile feedback may take several forms: attractive or repulsive forces [37,31], vibrations [46,44], temperature [23,10], pressure [19], electric current [25,28,43] or a combination of these parameters [14,58]. Tactile guidance may rely on simple feedback or sequential patterns [18,32] and can take place anywhere on the body (belt [18], wrist [21], back [44,54]...). On this subject, Cholewiak et al. suggest that anatomical reference points are more effective for tactile feedback perception [6].

Physicality

Physicality is an orthogonal dimension in regard to modality. Physical objects intrinsically exist in 3D space and can be directly perceived through the visual, audio and tactile modalities. Nonetheless, the fact that an object is physical can alter the nature of the object to such a degree that we consider its physicality to be paramount to the particular senses involved.

Physicality may facilitate the interpretation and understanding of information conveyed using virtual representations of the same modalities.

Guidance cues for orientation on the Vertical plan	Tactile Type: Relative	[32]							
	Tactile Type: Absolute	[14]							
	Auditory Type: Relative	1D sound [39] Big left/right							
	Auditory Type: Absolute	[45, 26]							
	Visual Type: Relative	2D Arrow							
	Visual Type: Absolute	Spot of light, 3D virtual arrow [33,13,16,41,36,5]	Projection Augmented Arm		[16]	[13]			
	Tangible Type: Relative								
	Tangible Type: Absolute	Post-it [38,29]							
	No Guidance		Compass Arm-pointing	[48,53,4]	2D Arrow [34,4,16]	[12,13,15]	[52,57]	[54]	[18] left/right Tap
	No Guidance	Tangible Type: Absolute	Tangible Type: Relative	Visual Type: Absolute	Visual Type: Relative	Auditory Type: Absolute	Auditory Type: Relative	Tactile Type: Absolute	Tactile Type: Relative

Guidance cues for orientation on the horizontal plan

Figure 2: The second representation of the design space, which respectively displays micro and macro guidance techniques on the vertical and horizontal axes. The “no guidance” column corresponds to micro-guidance techniques that provide no macro-guidance (and vice-versa for the “no guidance” row). Most techniques tend to focus on either macro (bottom row) or micro (left axis) navigation. Techniques in the middle combine micro and macro granularities. The table highlights that many combinations have not yet been explored so far.

For example, Kuzuoka et al. observe that the physical presence of a robot instructor makes the user aware of its movement and that the user could use this information to predict the location of the target pointed by the robot [29]. In another study, Mikawa et al. found that using a physical object (the head of a robot) to guide a library user would help him to understand guidance information [38]. Finally, Ju and Sirkin have shown that using a physical arm could be effective to attract the user’s attention [56].

Physical visual guidance use movement, shape and the state of physical objects to specify a target as in Kuzuoka et al.’s work [29]. In contrast, *virtual visual guidance* means that synthesized representations are used, as in [16] where a 3D arrow pointing to a target is displayed on a head mounted display.

Physical auditory guidance, which is emitted directly by physical objects [26], provides information on their 3D location and changes of state (e.g. the object is moving, stopped, collided...). *Virtual auditory guidance* relies on computer generated sounds that are emitted virtually in real space. Virtual sound can be played on headphones using head-related transfer functions [30,12] or by loudspeakers [45].

Physical tactile guidance results from touching physical objects and combines information on shape, changes in shape, acceleration, vibrations and temperature. *Virtual tactile guidance* requires creating illusions to provide feedback during interaction with virtual objects by using a cross-modal effect or by simulating tactile feedback using electrical [25,43], thermal or force [14]. For example, Ban et al.

propose a cross-model effect (combining visual and static tactile feedback) to create the illusion of tactile perception of the shape of virtual objects [3]. An example of simulated tactile feedback is the display proposed by Kajimoto et al. which simulates uses electrical stimulation to simulate tactile feedback (the sensation of pressure and vibration) [25].

In order to simplify the representation of the design space, Figures 1 and 2 project modality and physicality onto a single axis.

Granularity

Search tasks involve various phases depending on the scale of the target and its distance from the user. Generally, this task can be broken down into two broad types: macro navigation and micro navigation [22,50].

In *macro navigation*, the user tends to perform large movements, re-orienting him- or herself and moving around within the environment [54,4,21]. In *micro navigation*, the user is generally in the proximity of the target and performs a finer grained search, such as by scanning the field of view with the eyes [41, 16], hands [14] or both [32].

A guidance task that is well suited to macro navigation may not be particularly well-suited to micro navigation and vice versa. For example, drawing a small circle around a target may be effective when near the target [41,39], but if that target is on the other side of the room, it may be too small to identify. Likewise, emitting a beep from a small switch may be sufficient to lead the user across the room to a switchboard, but may be insufficient to identify the particular switch.

Spatial Guidance Type

Exocentric. These techniques display information in fixed locations or toward a fixed direction in space. As such, when the user moves or turns his or her body, the guidance information continues to present the same information. Examples of this type of technique include a mini-map in a videogame, a map [34, 24] (Figure 3c); a spotlight on the target [5,41] (Figure 3b); modifying the target [53] or a beep emitted from the target [26].

Egocentric. These techniques display information that is directly related to the user's location in the physical space. The information may be a sequence of instructions that the user needs to follow iteratively. Examples include projected arrows pointing to the left and to the right [34] (Figure 3a); text [40]; left/right beeps on headphones [57] or loudspeakers [39]; left/right tap on tactile belt [18], or a tap pattern with tactile feedback [9].

DISCUSSION

We provide two primary representations of the design space, as shown in Figures 1 and 2. Both of these representations provide a two-dimensional visualization of the proposed four-dimensional design space. Figure 1 shows modality and physicality vs. granularity and spatial representation type. Figure 2 focuses on the granularity of the guidance technique.

Orienting the second representation around the granularity of the guidance technique makes it easier to compare hybrid guidance techniques. These techniques are often used to provide one kind of guidance for the macro phase of the search task and another for the micro phase of the search task, as in Grohn et al. [13], who use 3D exocentric audio for coarse guidance, reinforced by a spotlight on the target (absolute visual guidance) for fine-grained guidance.

Together, these representations help show the descriptive, comparative, and generative capabilities of this design space.

Descriptive power. The proposed design space breaks guidance techniques down across the modality used to guide the user, the relationship between the user and the target, and the granularity of the search task.

This breakdown makes the proposed design space particularly good at describing hybrid guidance techniques, especially when using the second representation, as shown in Figure 2.

The majority of guidance techniques tend to focus on either macro or micro navigation. For example, all the techniques along the bottom row of Figure 2 provide macro guidance but

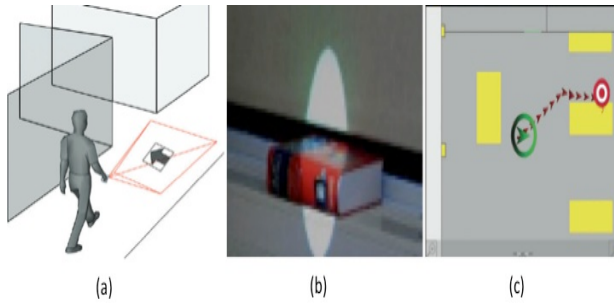


Figure 3 (a): Projected left/right arrow [34]. (b): A spotlight on the target [5]. (c): A map with the path [24].

no support for fine-grained micro navigation.

Along the left axis, techniques provide micro-navigation support, but do not target coarser macro-navigation. Techniques in the middle, however, combine modalities to support both granularities, enabling a broad description of even hybrid guidance techniques.

For example, Löchtfeld et al.'s Shelftorchlight [36], which displays a projected spotlight (exocentric visual micro guidance) on the target, is beyond the scope of Buchmann et al.'s classification, which focuses on macro guidance techniques and does not take into account the difference between modalities.

Comparative power. The representation of the design space shown in Figure 1 describes techniques in terms of modality, physicality, type, and granularity in a way that helps to identify techniques that are compatible for comparison. To compare two techniques, it is important to have only one parameter that varies. When several parameters vary, it is hard to link the difference in outcomes to a specific parameter.

The first representation classifies techniques in a way that each two horizontally adjacent techniques and techniques in the same column have only one distinct parameter (the modality, type or granularity). As such, for two techniques to be comparable, they should be horizontally adjacent or in the same column.

This representation reveals, for example, that Grohn et al.'s comparison of a 3D audio macro guidance technique to an exocentric visual technique for micro navigation (finding that the

audio technique is more effective to orient the user toward the direction of the target compared to the visual condition) [13] varies several parameters simultaneously. It is thus difficult to draw more general conclusions from their results.

This representation, however, only applies to non-hybrid techniques. More specifically, for hybrid techniques, it is necessary to break the individual sub-techniques into their individual components.

Generative power. The generative power of a design space describes the ability to create novel techniques. As before, Figure 1 reveals no guidance system with physical characteristics to support macro navigation. Figure 2 reveals that few guidance techniques apply to both macro and micro search tasks. Furthermore, it provides a suggestion of ways to create new hybrid techniques by blending techniques along the axes into a hybrid technique.

For example, we have used the representation in Figure 2 to identify a new technique, which we call an augmented projection arm, highlighted in yellow in the figure. It combines a physical macro exocentric guidance technique with a micro exocentric visual guidance technique. This prototype is currently under development, and we hope to be able to describe it in future work.

CONCLUSION

In this work we propose four dimensions to classify guidance techniques: modality, physicality, granularity of the search task and the spatial type of guidance. We then present two complementary representations of the design space. We further discuss its descriptive, comparative and generative power.

Our classification focuses on guidance techniques for the localization of real objects in 3D real space. It would be interesting, however, to explore the interactions between guidance techniques and localization technology constraints (e.g. precision, momentary failures, blind spots, etc.).

Our design space is also independent of the technology used to display the information. Future work should take into account the specificities of output devices such as augmented reality glasses.

ACKNOWLEDGMENTS

This work is funded by Cluster CONNEXION, a French project, and we would like to take this opportunity to thank Catherine Devic, François Cheriaux, Genevieve Filippi, Bernard Nouailhas from EDF and the others collaborators.

REFERENCES

- Alastair, D. S., Iain, D. G., and Bruce, M. H. Children's search behavior in large-scale space: Developmental components of exploration. In *Journal of Perception*, Pion Ltd, 34(10), 2005, pp.1221–1229.
- Bailey, S. W., Pampel, F. S., Ashmead, D., and Bodenheimer, B. Spatial localization with only auditory cues: a preliminary study. In *Proceedings of Applied Perception in Graphics and Visualization '11*, ACM, pp. 124–124.
- Ban, Y., Narumi, T., Tanikawa, T., and Hirose, M. Magic pot: Interactive metamorphosis of the perceived shape. In *SIGGRAPH '12*, ACM, pp. 1–1.
- Buchmann, V., Billingham, M., and Cockburn, A. Directional interfaces for wearable augmented reality. In *SIGCHI New Zealand '08*, ACM, pp. 47–54.
- Butz, A., Schneider, M., and Spassova, M. Searchlight – a lightweight search function for pervasive environments. In *PERVASIVE '04*, Springer Berlin Heidelberg, pp. 351–356.
- Cholewiak, R. W., and Criag, J. Vibrotactile pattern recognition and discrimination at several body sites. In *Journal of Perception & Psychophysics '84*, Springer-Verlag, pp. 503–515.
- Danieau, F., Lecuyer, A., Guillotel, P., Fleureau, J., Mollet, N., and Christie, M. Enhancing audiovisual experience with haptic feedback: A survey on hav. In *HAPTICS '13*, IEEE, pp. 193–205.
- El-Shimy, D., Marentakis, G., and Cooperstock, J. R. Tech-note: Multimodal feedback in 3d target acquisition. In *3DUI '09*, IEEE, pp. 95–98.
- Ertan, S., Lee, C., Willets, A., Tan, H., and Pentland, A. A wearable haptic navigation guidance system. In *International Symposium on Wearable Computers '98*, IEEE, pp. 164–165.
- Gallo, S., Cucu, L., Thevenaz, N., Sengul, A., and Bleuler, H. Design and control of a novel thermo-tactile multimodal display. In *HAPTICS '14*, IEEE, 75–81.
- Gaver, W. W. Synthesizing auditory icons. In *CHI '93*, ACM, pp. 228–235.
- Gonot, A., Chateau, N., and Emerit, M. Usability of 3D-Sound for Navigation in a Constrained Virtual Environment. In the 120th Audio Engineering Society Convention, Paris, 2006, pp. 1-16.
- Gröhn, M., Lokki, T., and Takala, T. Comparison of auditory, visual, and audiovisual navigation in a 3d space. In *Transactions on Applied Perception '05*, ACM, pp. 564–570.
- Guiatni, M., Riboulet, V., Duriez, C., Kheddar, A., and Cotin, S. A combined force and thermal feedback interface for minimally invasive procedures simulation. In *Journal of Mechatronics, IEEE/ASME '13*, 18(3), pp. 1170–1181.
- Harada, S., Takagi, H., and Asakawa, C. On the audio representation of radial direction. In *CHI '11*, ACM, pp. 2779–2788.
- Henderson, S., and Feiner, S. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *ISMAR '09*, IEEE, pp. 135–144.
- Hermann, T., Hunt, A., and John, G. N. *The Sonification Handbook*. Logos Publishing House, 2011.
- Hoggan, E., and Brewster, S. Crossmodal spatial location: initial experiments. In *NordiCHI '06*, ACM, pp. 469–472.
- Hoshi, T., Takahashi, M., Iwamoto, T., and Shinoda, H. Noncontact tactile display based on radiation pressure of airborne ultrasound. In *Haptics '10*, IEEE Transaction, pp. 155–165.
- Hsia, H. The information processing capacity of modality and channel performance. In *Journal of AV communication review'71*, 19(1), Springer US, pp. 51–75.
- Huxtable, B. J., Lai, C. K.-H., Zhu, J. W. J., Lam, P. M.-Y., Choi, Y. T., Neustaedter, C., and Corness, G. J. Ziklo: Bicycle navigation through tactile feedback. In *CHI EA '14*, ACM, pp. 177–178.
- Jacobson, R.D. Talking tactile maps and environmental audio beacons: An orientation and mobility development tool for visually impaired people. In *Proceedings of the ICA Commission on maps and graphics for blind and visually impaired people' 96*, pp. 1-22.
- Jones, L., and Berris, M. The psychophysics of temperature perception and thermal-interface design. In *Jouranal of HAPTICS's '12*, IEEE, pp. 137–142.
- Kahl, G., Spassova, L., Schöning, J., Gehring, S., and Kruüger, A. Irl smartcart - a user-adaptive context-aware interface for shopping assistance. In *Proceedings of Intelligent user interfaces '11*, ACM, pp. 359–362.
- Kajimoto, H., Kawakami, N., Maeda, T., and Tachi, S. Tactile feeling display using functional electrical stimulation. In *International Conference on Artificial Reality and Telexistence '99*, Virtual Reality Soc. of Japan, pp. 259–286.
- Kientz, J. A., Patel, S. N., Tyebkhan, A. Z., Gane, B., Wiley, J., and Abowd, G. D. Where's my stuff?: design and evaluation of a mobile system for locating lost items for the visually impaired. In *Proceedings of Assets '06*, ACM, pp. 103–110.
- Kortum, P. *HCI Beyond the GUI: Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.
- Kruijff, E., Schmalstieg, D., and Beckhaus, S. Using neuromuscular electrical stimulation for pseudo-haptic

- feedback. In *Proceedings of Virtual reality software and technology '06*, ACM, pp. 316–319.
29. Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., and Mitsuishi, M. Gestureman: A mobile robot that embodies a remote instructor's actions. In *Proceedings of Computer Supported Cooperative Work '00*, ACM, pp. 155–162.
 30. Larsen, C. H., Lauritsen, D. S., Larsen, J. J., Pilgaard, M., and Madsen, J. B. Differences in human audio localization performance between a hrtf- and a non-hrtf audio system. In *Proceedings of Audio Mostly '13*, ACM, pp. 1–8.
 31. Lecuyer, A., Coquillart, S., Kheddar, A., Richard, P., and Coiffet, P. Pseudo-haptic feedback: Can isometric input devices simulate force feedback. In *Proceedings of Virtual Reality '00*, IEEE, pp. 83–90.
 32. Lehtinen, V., Oulasvirta, A., Salovaara, A., and Nurmi, P. Dynamic tactile guidance for visual search tasks. In *UIST '12*, ACM, pp. 445–452.
 33. Li, M., Arning, K., Bremen, L., Sack, O., Ziefle, M., and Kobbelt, L. Profi: design and evaluation of a product finder in a supermarket scenario. In *Proceedings of UbiComp '13*, ACM, pp. 977–984.
 34. Li, M., Arning, K., Sack, O., Park, J., Kim, M.-H., Ziefle, M., and Kobbelt, L. Evaluation of a mobile projector-based indoor navigation interface. In *Journal Interacting with Computers '13*, 26(5), pp. 1-19.
 35. Lindeman, R. W., and Noma, H. A classification scheme for multi-sensory augmented reality. In *Proceedings of Virtual reality software and technology '07*, ACM, 175–178.
 36. Löchtefeld, M., Gehring, S., Schöning, J., and Krüger, A. Shelftorchlight: Augmenting a shelf using a camera projector unit. In *Conference on Pervasive Computing '10*.
 37. Menelas, B., Picinalli, L., Katz, B., and Bourdot, P. Audio haptic feedbacks for an acquisition task in a multi-target context. In *Symposium on 3DUI '10*, IEEE, pp.51–54.
 38. Mikawa, M., Morimoto, Y., and Tanaka, K. Guidance method using laser pointer and gestures for librarian robot. In *Conference of RO-MAN '10*, IEEE, pp. 373–378.
 39. Ngo, M., and Spence, C. Auditory, tactile, and multisensory cues facilitate search for dynamic visual stimuli. In *Journal of Attention, Perception, and Psychophysics '10*, 72(6), Springer-Verlag, 1654–1665.
 40. Nickels, J., Knierim, P., Ko'nings, B., Schaub, F., Wiedersheim, B., Musiol, S., and Weber, M. Find my stuff: Supporting physical objects search with relative positioning. In *Proceedings of UbiComp '13*, ACM, pp. 325–334.
 41. Ogata, K., Seya, Y., Watanabe, K., and Ifukube, T. Effects of visual cues on the complicated search task. In *NordiCHI '12*, ACM, pp. 478–485.
 42. Parsehian, G., and Katz, B. F. G. Morphocons: A new sonification concept based on morphological earcons. In *Journal of Audio Eng. Soc '12*, 60(6), pp. 409–418.
 43. Pfeiffer, M., Schneegass, S., Alt, F., and Rohs, M. Let me grab this: A comparison of ems and vibration for haptic feedback in free-hand interaction. In *AH '14*, ACM, pp. 48:1–48:8.
 44. Prasad, M., Taele, P., Goldberg, D., and Hammond, T. A. Haptimoto: Turn-by-turn haptic route guidance interface for motorcyclists. In *CHI '14*, ACM, pp. 3597–3606.
 45. Pulkki, V. Virtual sound source positioning using vector base amplitude panning. In *Journal of Audio Eng. Soc '97*, 45(6), pp. 456–466.
 46. Salzer, Y., Oron-Gilad, T., and Ronen, A. Vibrotactor-belt on the thigh: Directions in the vertical plane. In *Proceedings of EuroHaptics '10*, Springer Berlin Heidelberg, pp. 359–364.
 47. Seco, F., Jimenez, A., Prieto, C., Roa, J., and Koutsou, K. A survey of mathematical methods for indoor localization. In *International Symposium on Intelligent Signal Processing' 09*, IEEE, pp. 9–14.
 48. Simon, R., Jennifer, P., and Jones, M. A billion signposts: Repurposing barcodes for indoor navigation. In *CHI '14*, ACM, pp. 639–642.
 49. Song, Z., Jiang, G., and Huang, C. A survey on indoor positioning technologies. In *International Conference Theoretical and Mathematical Foundations of Computer Science '11*. Springer Berlin Heidelberg, pp. 198–206.
 50. Stahl, C., Baus, J., Brandherm, B., Schmitz, M., and Schwartz, T. Navigational- and shopping assistance on the basis of user interactions in intelligent environments. In *The IEE International Workshop on Intelligent Environments '05*, IEE, pp. 182–191.
 51. Stent, A. J., Azenkot, S., and Stern, B. Iwalk: a lightweight navigation system for low-vision users. In *Proceedings of ASSETS '10*, ACM, pp. 269–270.
 52. Takase, Y., and Hasegawa, S. Presentation of directional information by sound field control. In *AH '12*, ACM, pp. 32:1–32:3.
 53. Takeuchi, Y., and Perlin, K. Clayvision: the (elastic) image of the city. In *CHI '12*, ACM, pp. 2411–2420.
 54. Tan, H. Z., Gray, R., Young, J. J., and Traylor, R. A haptic back display for attentional and directional cueing. In *Journal of Haptics Research '03*, 3(1), pp. 1-20.
 55. van Krevelen, D., and Poelman, R. A survey of augmented reality technologies, applications and limitations. In *The International Journal of Virtual Reality '10*, 9(2), pp. 1-20.
 56. Wendy, J., and David, S. Animate objects: How physical motion encourages public interaction. In *Proceedings of Persuasive Technology '10*, Springer Berlin Heidelberg, pp. 40–51.
 57. Yamano, S., Hamajo, T., Takahashi, S., and Higuchi, K. Eyesound: single-modal mobile navigation using directionally annotated music. In *AH '12*, ACM, pp. 1–4.
 58. Yang, G.-H., Kyung, K.-U., Srinivasan, M., and Kwon, D.-S. Quantitative tactile display device with pin-array type tactile feedback and thermal feedback. In *International Conference on Robotics and Automation '06*, IEEE, pp. 3917–3922.

Interactions gestuelles pour Strip'TIC, un environnement tangible pour les contrôleurs aériens

Yoann Gauthier¹Joran Marcy¹David Duprat¹Alexis Paoleschi¹

{yoann.gauthier9, joran.marcy, dprt.david, alexis.paoleschi}@gmail.com

Catherine Letondal¹Rémi Lesbordes²Jean-Luc Vinot¹Christophe Hurter¹

¹Université de Toulouse/ENAC
31055, Toulouse, France
<prénom>.<nom>@enac.fr

²DTI R&D DGAC DSNA,
Toulouse, France
remi.lesbordes@aviation-civile.gouv.fr

RESUME

Dans cet article nous décrivons une exploration des possibilités ouvertes par l'interaction gestuelle dans un système interactif mixte pour les contrôleurs aériens. Cette exploration repose sur une analyse des gestes des contrôleurs que nous avons pu observer en tour de contrôle et en simulateur. La conception exploratoire que nous proposons nous permet en particulier d'évaluer la pertinence de l'interaction gestuelle pour les objets virtuels associés aux objets physiques.

Mots Clés

Interaction gestuelle ; interaction tangible ; conception participative ; ethnographie ; interaction avec le papier ; contrôle aérien.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Une modalité d'entrée particulière, qu'il s'agisse du contact avec une surface tactile, du déplacement d'un objet capturé par l'usage de tags de réalité virtuelle, de la détection d'un mouvement d'une partie du corps dans l'espace, ou d'un stylo numérique, contraint à sa manière les possibilités d'interaction dans un environnement interactif. L'interaction tactile oblige à un contact avec la surface, la manipulation d'un objet détectable ne permet pas d'interaction fine avec cet objet, l'utilisation d'un stylo mobilise les doigts, et la détection d'un mouvement ou d'une posture corporelle renforce la tendance au geste sémiotique, parfois artificiel. Dans le domaine du contrôle aérien, les interactions très riches des contrôleurs avec leurs outils offrent un terrain intéressant pour explorer ces contraintes. Pour assurer la régulation du trafic et la surveillance des vols, les contrôleurs manipulent en effet des strips, bandes de papier correspondant à un avion contrôlé avec ses informations

de vols. Les strips sont déplacés, annotés et organisés sur un tableau de strips, désignés et transmis entre contrôleurs, stockés pour archivage... et ces manipulations peuvent se décrire à travers les gestes qui leur sont associés.

Cette étude s'intègre dans le cadre du projet Strip'TIC [15], un système mixte de recherche pour le contrôle aérien. Strip'TIC dispose de plusieurs modalités d'entrée et supporte non seulement l'utilisation de strips papier traditionnels, augmentés par le système, mais également leurs doubles électroniques, appelés strips virtuels, dont la représentation est projetée sous le strip papier. Les modalités proposées sont notamment le multi-touch pour interagir avec les strips virtuels et le tableau de strips [24], la détection des strips papier sur le tableau par l'intermédiaire de codes de réalité augmentée, et le stylo numérique, non seulement pour écrire des instructions ou sélectionner des zones sur le strip, mais aussi pour servir de doigt interagissant avec les informations affichées.

Dans cette étude, nous nous proposons d'explorer l'utilisation de techniques de détection de gestes afin d'évaluer leur potentiel. Notre objectif, à partir d'une analyse des gestes actuels des contrôleurs aériens, a été de conduire une démarche participative avec les contrôleurs dans le but de concevoir et d'implémenter dans Strip'TIC un ensemble de gestes adaptés à l'activité, et permettant notamment de mieux harmoniser les manipulations des strips physiques et des strips virtuels.

Après une présentation du contrôle aérien, l'article présente nos observations des gestes de l'activité, nos explorations pour le design d'interactions gestuelles adaptées et leur implémentation dans le prototype Strip'TIC. Nous concluons par une discussion sur notre contribution et une ouverture sur les travaux futurs.

ETAT DE L'ART

La modalité gestuelle, comme dans l'exemple du clavier gestuel [28] qui permet d'augmenter un clavier mécanique avec des gestes pour faciliter la navigation, est une modalité qui peut compléter et enrichir l'interaction avec un système. Strip'TIC étant composé d'une table multi-touch, nous focaliserons notre étude sur les interactions gestuelles sur surface interactive.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670457>

L'utilisation classique de ces tables s'effectue avec des gestes en deux dimensions mais plusieurs études ont montré l'utilité des interactions gestuelles en trois dimensions [3, 8, 10, 18 30]. Utiliser la gestuelle 3D au-dessus de la table interactive vise notamment à compléter l'interaction tactile, tout en restant dans le même espace d'interaction. Marquardt et al. [20] montrent qu'il est possible de créer des nouvelles interactions en unifiant à la fois le tactile et la gestuelle ; on parle d'interaction continue dans l'espace. Hilliges et al. [12] mettent en relief les problèmes d'apprentissage et de non-intuitivité de ces interactions continues. [14] propose d'utiliser la familiarité avec les outils physiques équivalents pour pallier ce problème.

Plusieurs techniques et systèmes ont été développés pour capter les gestes des utilisateurs. On peut notamment citer les gants munis de capteurs de position (DataGlove [31]), les caméras stéréoscopiques, les caméras avec détection du relief (e.g. Kinect de Microsoft), les caméras classiques ou webcams (Shimada et al. [25]). SkinPut [13] utilise le corps humain comme un outil de transmission acoustique, l'analyse des vibrations mécaniques lors du touché permet de détecter les entrées. Excepté le dispositif de traitement, l'outil de détection est transparent pour l'utilisateur. L'environnement peut être aussi utilisé comme le système Humantenna [6] qui reconnaît les gestes du corps humain grâce aux perturbations d'un signal électromagnétique engendrés par les mouvements du corps. SoundWave [9] est capable de reconnaître les mouvements d'une main au-dessus d'une surface en enregistrant les variations de fréquence d'un son émis à haute fréquence.

Les feedbacks et les feedforwards sont primordiaux pour informer l'utilisateur du niveau de compréhension de ses gestes par le système [29]. GIDeS [22] donne à l'utilisateur des informations contextuelles en fonction du geste reconnu pour la conception assistée par ordinateur. L'utilisation d'ombre pour indiquer à l'utilisateur le déplacement d'un objet virtuel permet de donner plus de réalisme à la manipulation et ainsi augmenter la naturalité [12]. Ce procédé a déjà été utilisé dans des systèmes de collaborations à distance pour représenter les actions des autres utilisateurs [11, 26].

LE CONTROLE AERIEN

L'activité des contrôleurs aériens consiste principalement à maintenir une distance de sécurité entre avions et à optimiser l'écoulement du trafic aérien (par exemple la minimisation des retards à l'atterrissage). Pour ce faire, les contrôleurs aériens supervisent les informations des vols traversant un secteur d'espace contrôlé et donnent des ordres (clairances radio) de cap, de vitesse ou d'altitude, aux avions. Dans un environnement de travail typique, deux contrôleurs sont assis devant une position de contrôle avec deux écrans radar présentant une image temps réel du trafic et un tableau horizontal commun servant de support physique à des bandes de papier, appelées strips, sur lesquelles sont imprimées diverses informations relatives

au vol et la route de l'avion, et sur lesquelles les contrôleurs notent les instructions données aux pilotes.

Un exemple : la gestion des stacks

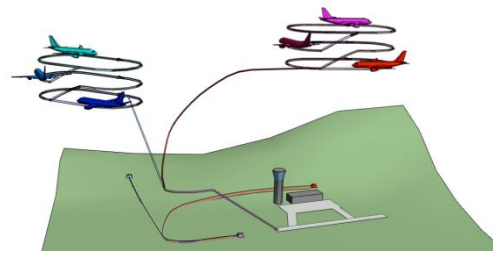


Figure 1: hippodrome (stack) d'attente des avions avant leur atterrissage sur une plateforme aéroportuaire

A titre d'exemple, la gestion des hippodromes d'attente (stack) est une activité courante des contrôleurs aériens d'approche (Figure 1). Les avions placés en attente réalisent des trajectoires en forme d'hippodromes, chaque avion avec une altitude différente. Lorsque les conditions d'atterrissage deviennent possibles (piste disponible), le contrôleur demande à l'avion ayant la plus faible altitude de sortir du stack pour atterrir. Cette configuration nécessite beaucoup de surveillance et d'ordres de clairance radio. Les situations peuvent aussi devenir complexes avec des conditions météorologiques dégradées ou des avions présentant des urgences (ex. un avion avec un niveau bas de carburant). La configuration physique des strips papier devient elle aussi plus complexe avec un arrangement spécifique sur le tableau de strips (Figure 2).

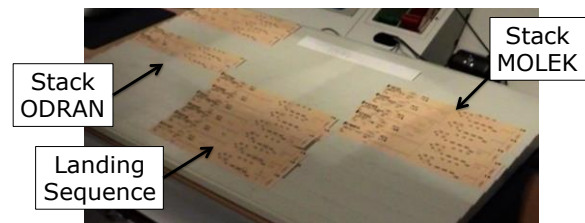


Figure 2: arrangement de strips papier sur le tableau de strip avec la répartition en hippodromes (stack) d'attente.

Le prototype Strip'TIC

Dans le cadre d'un projet de recherche, cette gestion de strips a été informatisée dans le projet Strip'TIC (Stripping Tangible Interface for Controllers) [15], avec pour premier objectif d'étudier l'utilisation du stylo électronique dans le domaine du contrôle aérien, puis de permettre une exploration du design d'interaction entre espaces physique et numérique [17]. Strip'TIC combine le stylo numérique Anoto, une dalle multi-touch [24], les technologies de réalité augmentée [15] pour détecter la position des strips sur le tableau, et la projection au dessus et en dessous de la dalle. Les strips papier, l'image radar et le tableau sont recouverts du motif Anoto. Les contrôleurs utilisent le stylo numérique pour annoter les strips (e.g. noter les instructions données aux pilotes), ils peuvent pointer sur des informations, comme les balises, ou sélectionner un niveau de vol (Figure 3). Le système est à chaque fois informé des tracés effectués et peut les traiter, comme pour de la reconnaissance d'écriture ou de la

sélection d'information. Le système projette sur les strips des informations dynamiques, relatives au vol ou répondant à des requêtes des contrôleurs (ex. Quels vols passent par cette balise, sélectionnée sur l'image radar ou sur l'un des strips). Strip'TIC est aujourd'hui un prototype fonctionnel, une plateforme de support à la recherche en interaction dans le domaine critique du contrôle aérien. Notre étude s'inscrit dans ce cadre de recherche, avec pour objectif d'investiguer l'apport de l'interaction gestuelle 3D au sein du dispositif Strip'TIC.



Figure 3: Dans Strip'TIC a) le contrôleur peut interagir avec le stylo Anoto sur l'image radar, les strips et le tableau, b) chaque strip papier est associé à un strip virtuel affiché en dessous, qui affiche les mêmes informations. Le système projette des informations dynamiques par dessus les strips physiques et virtuels, ainsi que sur le tableau.

METHODOLOGIE

Cette étude s'est déroulée en deux phases principales, distantes d'une année. Une première phase qui impliquait 2 concepteurs d'interaction et 1 contrôleur aérien, a consisté en 2 séances d'observation de 1h et 2h de contrôleurs à l'aéroport de Blagnac à Toulouse, dans la salle de contrôle en approche et dans la tour, complétées par de brefs interviews avec les contrôleurs disponibles. Elle a été enregistrée (vidéo et Livescribe) et transcrite, en se focalisant plus particulièrement sur les gestes, leur contexte et les paroles associées. Cette première étude a fait l'objet d'une présentation orale [18].

La seconde phase s'est déroulée lors d'un chef d'œuvre d'étudiants du Master IHM de Toulouse, et a duré 6 mois. Durant ce projet, qui impliquait les concepteurs de Strip'TIC dont un contrôleur aérien comme encadrants, les étudiants ont effectué 2h d'observation aux simulateurs de formation des contrôleurs de l'ENAC. Des analyses précises des gestes ont été réalisées lors d'une seconde séance de 2h, sur la base d'une grille d'observation pré-établie. La conception des interactions gestuelles qui a été réalisée au cours de ce projet a impliqué 2 ateliers participatifs (brainstorming, prototypage vidéo) avec des contrôleurs (2-3), des concepteurs d'interaction (6) et des ergonomes (1-2). Deux séances d'évaluation ont été réalisées avec des spécialistes du domaine.

ETUDE DE L'ACTIVITE AUTOUR DES GESTES AVEC DES STRIPS PAPIERS

Cette partie décrit certains gestes régulièrement utilisés par les contrôleurs aériens durant leur activité, gestes que nous avons pu observer et analyser lors des deux phases

de l'étude. Cette analyse, à la différence d'une étude telle que [19] portant sur l'ensemble de l'activité des contrôleurs, s'est focalisée uniquement sur la partie gestuelle. Les gestes observés peuvent être décrits à trois niveaux : un niveau général proposant une typologie et un cadre global à l'analyse des gestes des contrôleurs, un niveau détaillé précisant les caractéristiques de gestes spécifiques, notamment ceux impliquant la manipulation des strips papier, et enfin un niveau scénario, qui décrit les manipulations des strips dans des situations métier caractéristiques.



Figure 4 : typologie des gestes observés. a) geste « beat » pour soutenir une description de trajectoire, b) saisie d'un strip de l'imprimante, c,d) bimanualité, e) geste déictique vers l'écran, f) écriture et désignation d'un strip, g) écriture avec maintien du strip par la main micro, h) geste épistémique pour réfléchir à une action potentielle sur 2 vols, i) plusieurs instruments dans la main, j) position de préparation/concentration, k) geste interactionnel pour indiquer une intention de communiquer, l) manipulation de plusieurs strips simultanément, m) léger soulèvement d'un strip pour désignation, n) pointage, o) double pointage.

Observations des gestes des contrôleurs

La première phase d'observation à Blagnac a permis de constater une grande diversité des gestes dans l'activité du contrôleur. La Figure 4 présente des images de ces gestes extraites des enregistrements vidéo.

Notre acception du terme « geste » est ici très inclusive. Elle comprend tout d'abord les gestes sémiotiques, essentiellement déictiques [23] comme désigner un strip (Fig. 4.f,n) voire deux strips, par exemple pour les comparer (Fig. 4.o : « ben c'est qui qui part devant ? »), ou des éléments de l'image radar (Fig. 4.e), les gestes ergotiques, essentiellement pour la manipulation et le déplacements de strips. Elle comporte également des gestes non sémiotiques, qu'on peut qualifier selon le contexte d'épistémiques (Fig. 4.h,j), interactionnels [21] (Fig. 4.k), irrelevant [21] comme les tics ou le fait de faire tourner son stylo, ou les « beat gestures » [5], mettant l'emphase sur certains points de leur discours (Fig. 4.a). Les gestes observés s'accompagnent ou non de l'utilisation d'un instrument (stylo, crayon, souris, micro). Une seule main peut souvent en tenir plusieurs, notamment stylo et micro (Fig. 4.i). Différents types de gestes bimanuels ont pu être notés, par exemple pour inverser des strips ou reclasser deux stacks d'attente en même temps (Fig. 4.c,d). Nous avons cependant remarqué qu'une seule main est utilisée pour les actions à contrôler séquentiellement ou lorsque le contrôleur doit tenir le téléphone. Concernant la manipulation de strips, on observe souvent qu'elle peut impliquer plusieurs strips en même temps (Fig. 4.c,d,h,l), conduisant parfois aussi à des regroupements de strips (empilement par exemple). Enfin les gestes portent fréquemment sur le tableau de strips de l'autre contrôleur.



Figure 5: Position des doigts pour saisir un strip en fonction du contexte. Orange : pour passer à un collègue, vert : saisie avec deux mains, violet : saisie pour déplacement.

Lors de la séance d'observation de 2h en salle de simulateur (1h de simulation et 1h de débriefing), nous avons pu observer des gestes particuliers, en suivant une grille d'observation et en nous répartissant l'observation des gestes entre 4 observateurs. La grille d'observation comportait les gestes de déplacement, pointage, saisie, maintien et écriture, ainsi que la position des contrôleurs, et avec pour chaque geste à observer les propriétés associées, telles que : ampleur, nombre de doigts, contexte, ou rôle de chaque main. Nous avons ainsi plus particulièrement analysé les gestes ergotiques ou déictiques associés à la manipulation des strips : le déplacement de strips (Fig. 4.b,c,d,h,l,m), le pointage d'un strip, la saisie, le maintien (Fig. 4.g) et l'écriture sur le strip. Notamment, nous avons pu déterminer la manière dont le strip est saisi en terme de positions des doigts, qui diffèrent selon le contexte (cf. Figure 5).

Des scénarios de manipulation gestuelle

Enfin, nous avons identifié avec les contrôleurs quatre scénarios de manipulation des strips, qui ont fourni un cadre pour la conception de gestes pour Strip'TIC.

Sc.1 : sortie d'un vol d'un stack. Cette opération est effectuée manuellement par le contrôleur radariste qui consiste à sortir un strip de sa colonne de stack (Figure 2) pour ensuite être pris en charge par un autre contrôleur et assurer son atterrissage.



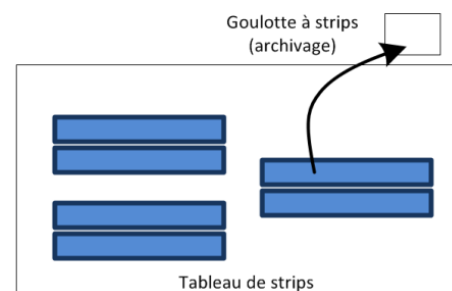
Lorsqu'un strip est supprimé de sa colonne, les strips sont réarrangés pour combler l'espace ainsi libéré (décalage de plusieurs strips vers le haut ou vers le bas).

Sc.2 : particularisation d'un avion. Lorsque le contrôleur souhaite particulariser un avion au travers de son strip, il utilise l'indentation (décalage horizontal du strip par rapport aux autres).



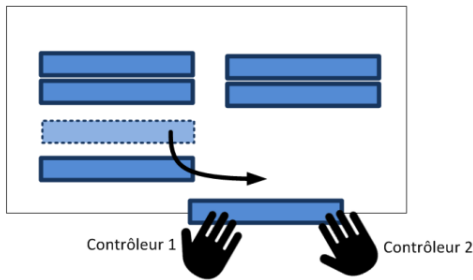
Cette action permet de marquer un vol et est un moyen efficace pour se rappeler d'une action à effectuer ultérieurement sur le vol indenté. En complément, le contrôleur peut aussi garder son index pointé sur le strip et s'assurer un retour visuel rapide sur le strip concerné.

Sc.3 : archivage d'un avion. Le contrôleur a pour tâche de gérer le cycle de vie d'un strip papier. Lorsqu'un avion sort d'un secteur aérien et qu'il n'est plus sous sa responsabilité, le contrôleur sort le strip du tableau et le dépose dans un conteneur à strips pour archivage. Il effectue alors un mouvement de cloche avec sa main vers la goulotte à strips située en haut du tableau.



Sc.4 : transfert d'un avion au secteur suivant. Les strips papiers peuvent être directement échangés entre contrôleurs aériens sur des positions différentes.

Ce transfert d'objets physiques au travers du geste de transmission d'un strip, donne au contrôleur recevant la gestion et la responsabilité du vol ainsi reçu. Dans une grande majorité des cas ce transfert s'effectue avec un accord oral préalable et des échanges verbaux additionnels (par exemple la transmission des caractéristiques du vol).



ETUDE DE L'ACTIVITE AUTOUR DES GESTES AVEC DES STRIPS PAPIERS ET ELECTRONIQUES

A l'issue de l'analyse de l'activité, nous avons organisé deux ateliers de conception participative (brainstorming, prototypage basse fidélité) avec pour but d'éliciter des scénarios d'utilisation de l'interaction gestuelle dans Strip'TIC. Notre but était d'étudier comment les gestes pouvaient être utilisés à la fois pour la manipulation des strips physiques et virtuels. Ces séances de brainstorming et de prototypage vidéo ont été réalisées avec des contrôleurs aériens, ergonomes, designers, ingénieurs et chercheurs. Au final, trois types d'interactions gestuelles ont été jugés pertinentes :

- déplacement de strips virtuels : ces gestes permettent de déplacer les strips dans l'espace dans le cadre d'une gestion standard du tableau de strips, complétant ainsi l'interaction multi-touch existante dans Strip'TIC ;
- copie de strips réels en strips virtuels : ces gestes servant à dupliquer un strip afin d'en avoir une copie virtuelle découlent de l'identification de nouveaux besoins, quand par exemple travailler avec des copies de strips peut servir à simuler des séquençement dans une zone dédiée du tableau de strips ;
- interaction avec l'écran radar, comme le pointage d'objets sur l'écran radar pour améliorer la communication entre contrôleurs (fig. 4.d) ou la possibilité de manipuler l'image radar (pan & zoom) ; bien que ces interactions soient jugées par les contrôleurs risquées et moins précises ou performantes que les interactions existantes, ils ont néanmoins trouvé pertinent d'implémenter de tels gestes pour illustrer leur potentiel.

Implications pour la conception

Les ateliers participatifs ont été permis d'aboutir à des idées d'interactions gestuelles qui ont été évaluées par l'équipe à travers deux aspects : l'analyse gestuelle préalable et la faisabilité technique. Cette méthode de sélection a permis de retenir onze idées, qui ont toutes été implémentées. Les résultats de l'analyse gestuelle décrits plus haut ont ainsi été utilisés lors de la phase de conception pour définir des interactions gestuelles sur les strips virtuels « en accord » avec celles existant sur les strips papiers, comme le propose [14]. On peut par exemple citer le geste d'archivage des strips virtuels implémenté dans Strip'TIC qui est un mouvement ample de cloche calqué sur ce qui se fait pour un strip papier, ou le geste de punaise qui rappelle les gestes de maintien de strips, par exemple lors de l'écriture. (figure 4.g), ou pour réserver une action à faire sur un strip. L'analyse

des gestes de l'activité a aussi permis de proposer de nouvelles interactions corrigeant des problèmes identifiés. On peut citer le pointage à distance de l'écran et la manipulation de l'image radar qui évitent désormais à l'utilisateur de se lever de son fauteuil pour pointer l'image radar ou l'agrandir.

INTERACTIONS GESTUELLES

Dans cette section, nous décrivons les interactions gestuelles que nous avons conçues et implémentées.

Description du système

Les interactions

Notre objectif a été d'intégrer des interactions gestuelles enrichissant les interactions multi-touch existantes via un dispositif de capture d'interaction gestuelle. Pour cela, nous avons développé un système qui permet de reconnaître un ensemble de gestes effectués par les contrôleurs aériens concernant les strips ou l'image radar. La reconnaissance de la posture et de la position des mains (et doigts) est utilisée pour permettre des interactions entre l'utilisateur et le système Strip'TIC. Les différentes interactions implémentées : déplacement, copie et archivage de strips, ainsi que manipulation et pointage dans l'image radar, sont détaillées plus loin.

Les feedbacks

L'ajout de feedback est essentiel dans les systèmes qui interprètent les mouvements de l'utilisateur : il permet à l'utilisateur de connaître l'état du système, de corriger le geste en cas de mauvaise détection, et tend à le rendre plus tolérant aux erreurs de détection [16]. Dans notre conception, nous distinguons deux types de feedback. Tout d'abord, il y a des feedbacks projetés par en-dessous sur le tableau de strip et les strips, comme par exemple le contour du strip virtuel qui change de couleur si une main s'en approche. Il s'agit donc de feedback sur les objets d'intérêt. L'autre type de feedback consiste en une projection d'icône sur la main de l'utilisateur avec le projecteur supérieur [13] (cf. Figure 6). La main étant un moyen de manipulation central pour le contrôleur aérien nous avons voulu y afficher les feedbacks. Dans notre système, les deux types de feedback sont combinés : projection sur la main et projection d'un strip virtuel ombragé sur le tableau. L'utilisateur sait donc à la fois qu'il manipule un strip virtuel (ombre sur le tableau) et avec quelle main (icône sur la main).

Les contraintes

La détection des gestes devait être précise, posséder un champ de détection qui s'étendait au minimum sur toute la surface du tableau de strips et posséder un temps de latence faible. Plusieurs technologies furent envisagées pour répondre à ces critères, en particulier le Leap Motion qui venait d'arriver sur le marché et la Kinect de Microsoft qui avait suscité un enthousiasme certain dans le monde de la recherche. L'étude de ces différentes technologies a fait apparaître un compromis entre la surface de détection d'un dispositif et sa précision. Par exemple, le Leap Motion est précise alors que son champ de détection est limité. Le volume de détection est défini

à partir de deux arcs de cercle de rayon 61cm, l'un d'angle 120° pour l'avant/arrière, et l'autre d'angle 150° pour la gauche/droite. Nous avons aussi prêté une attention particulière aux kits de développement qui accompagnaient les dispositifs de reconnaissance. Le Leap Motion ne proposait que le SDK officiel alors qu'il existait de nombreux outils de développement pour la Kinect. De plus, le système Strip'TIC imposait des contraintes physiques pour l'intégration des dispositifs de reconnaissances, en particulier pour la Kinect qui ne pouvait pas être placée devant l'utilisateur dans des conditions suffisantes pour la détection. Ces contraintes furent prises en compte dans la recherche d'une solution complète pour la reconnaissance d'interactions gestuelles.

Les interactions

Les interactions retenues permettent d'enrichir les interactions avec les strips virtuels par du geste, dupliquer des strips virtuels, interagir à distance avec le système.

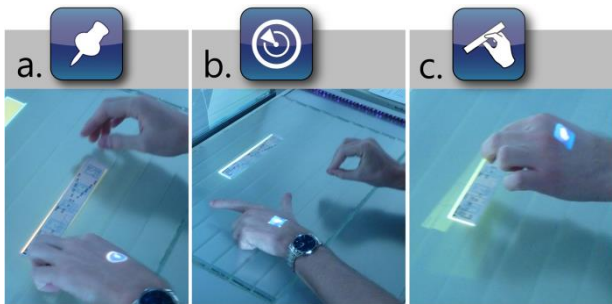


Figure 6 : Feedbacks projetés sur la main de l'opérateur : (de gauche à droite) : a) punaise, b) radar, c) saisie.

Enrichir les interactions avec les strips virtuels

La manipulation des strips virtuels s'effectue de la même manière que si l'utilisateur interagissait avec un strip papier, bien qu'il ait fallu modifier très légèrement certains gestes à cause de la mauvaise détection du dispositif de reconnaissance gestuelle. L'utilisateur approche sa main d'un strip virtuel, le bord du strip passe en vert dès lors qu'il touche la table. Il peut donc le saisir en effectuant un geste de pinch au contact de la table. Le système projette une icône de pinch sur la main de l'utilisateur pour l'informer qu'il a bien saisi le strip virtuel (.c). Le strip virtuel ainsi que la projection de l'icône suivent la main ayant réalisé la saisie du strip virtuel. La projection de l'icône disparaît quand le déplacement du strip est fini. L'interaction peut aboutir à deux résultats différents :

- l'utilisateur ouvre la main alors qu'il n'est pas en contact avec la table, cela annule le déplacement du strip qui revient à sa position initiale ;
- l'utilisateur ouvre la main alors qu'il est en contact avec la table, cela valide donc le déplacement du strip qui se fige à sa nouvelle position.

Le contrôleur doit pouvoir archiver les strips virtuels « orphelins » (ceux qui ne sont pas associés à un strip papier). La fonctionnalité de l'archivage combine la

reconnaissance par la Kinect et par le Leap Motion. L'archivage d'un strip virtuel débute comme le déplacement d'un strip virtuel. La variante arrive quand l'utilisateur sort la main ayant effectué le pinch du tableau de strip (c'est la Kinect qui permet de détecter que la main de l'utilisateur quitte le tableau de strips). Cette sortie a plusieurs conséquences tout d'abord un feedback est projeté sur la table pour informer l'utilisateur qu'il entre dans la zone d'archivage. Pour archiver le strip, l'utilisateur doit juste ouvrir sa main au-dessus du Leap Motion. Dès lors que l'archivage est effectué, le feedback projeté sur la table change de couleur et disparaît pour informer l'utilisateur que son action a bien été prise en compte. Dans le cas où l'utilisateur n'ouvre pas sa main dans la zone d'archivage et revient sur le tableau de strip, le strip virtuel repasse dans le mode déplacement. Bien qu'un strip soit archivé, il peut arriver que le contrôleur ait besoin de le remettre sur son tableau de strips, par exemple si le pilote de l'aéronef a mal compris la fréquence radio du secteur suivant. Pour réaliser cette action avec des strips virtuels sur le système Strip'TIC, l'utilisateur doit entrer une de ces mains dans la zone d'archivage, c'est-à-dire au-dessus du Leap Motion. Le système affiche alors la liste des derniers strips virtuels archivés, l'utilisateur en sélectionne un en déplaçant sa main au-dessus du Leap Motion vers le haut ou le bas. Un mouvement vers le haut permet de sélectionner un item plus haut dans la liste et vers le bas le contraire. Quand le strip est sélectionné, l'utilisateur ferme sa main toujours au-dessus du Leap Motion pour valider le désarchivage. Le retour de la main sur le tableau de strip permet le passage dans le mode déplacement du strip virtuel. L'utilisateur peut donc placer où il le souhaite le strip virtuel qu'il vient de désarchiver.

La duplication de strip peut être intéressante dans les situations où on retire le strip du tableau, comme c'est le cas lors d'un dégroupement (i.e. lorsque le trafic est réparti sur plusieurs positions de contrôle) ou pour des tests de séquençements (What if) dans le contrôle d'approche. Strip'TIC offre à l'utilisateur la possibilité de dupliquer les strips papiers et les strips virtuels. A partir d'un strip papier, il existe deux interactions de duplication en strip virtuel.

La première, intitulée "la peau" (Figure 7), consiste à retirer le pendant virtuel d'un strip physique pour le recopier ailleurs. Pour ce faire l'utilisateur s'approche du strip papier et effectue un mouvement de pinch au contact de celui-ci. Le système projette une icône de "peau" sur la main ayant réalisé le pinch et il y a création d'un strip virtuel qui est en mode déplacement. L'interaction peut se terminer de deux façons :

- l'utilisateur ouvre sa main sans toucher la table, la duplication est annulée ;
- l'utilisateur ouvre sa main au contact de la table, la duplication est validée et le strip reste à sa nouvelle position.

Duplication de strips virtuels

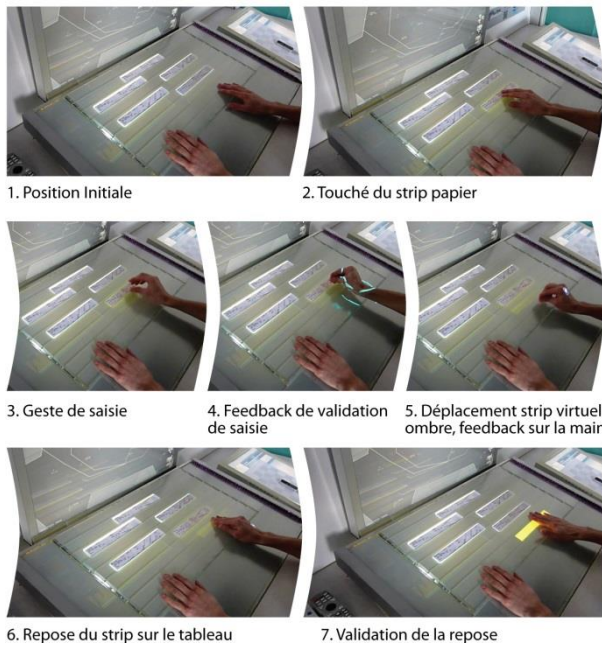


Figure 7 : Interactions gestuelles : la peau

La deuxième interaction est “la punaise”, qui consiste à punaiser le pendant virtuel d’un strip sur le tableau lors du retrait du strip papier. Il s’agit d’une interaction bi-manuelle. L’utilisateur touche avec l’index le strip à dupliquer. Le système affiche une icône de “punaise” sur la main. L’autre main est ensuite utilisée pour tirer le strip papier. La duplication est donc faite puisque si le strip papier est posé ailleurs, un nouveau pendant virtuel sera affiché. L’icône de punaise disparaît quand l’utilisateur arrête de toucher le strip avec l’index.

Concernant les strips virtuels (orphelins), l’interaction “punaise” a été reprise. Il s’agit à nouveau de toucher avec l’index le strip à dupliquer et effectuer un geste de pinch avec l’autre main au contact du strip. Le système projette une icône de “punaise” sur la main pointant le strip et une icône de “pinch” sur la main déplaçant le strip. La duplication est donc réalisée et le nouveau strip passe en mode déplacement. L’annulation s’effectue de la même façon que pour l’interaction “la peau”.

Interactions gestuelles à distance

Nous avons développé deux interactions gestuelles à distance permettant le pointage et le zoom de l’image radar.

Pointage : il est activé dès lors que l’utilisateur pointe avec son index l’écran radar. Cela a pour effet d’afficher un point rouge sur l’écran. Le point se déplace en fonction des mouvements de l’index de l’utilisateur. Pour sortir de ce mode, l’utilisateur n’a qu’à ouvrir la main. Cette interaction permet au contrôleur aérien de pointer l’écran radar sans se lever de son siège, notamment lors d’une discussion avec un collègue.

Pan & Zoom : l’utilisateur peut aussi régler l’échelle de la carte radar à distance. Il s’agit d’une interaction bi-

manuelle, avec une main l’utilisateur effectue un geste de L sur la table. Cela a pour effet la projection d’une icône radar sur la main concernée, l’autre main effectue un geste de pinch dans l’espace ce qui permet de saisir virtuellement l’écran radar. Le mouvement de rapprochement ou d’éloignement de la main permet le zoom et le dé-zoom, les mouvements dans les deux autres directions permettent le pan de la carte. Pour finaliser l’interaction l’utilisateur peut relâcher la carte (dans ce cas il reste en mode L) et peut donc la ressaisir à tout moment. Ou il peut aussi ouvrir la main ayant effectué le L, ce qui arrête toute manipulation avec l’écran radar.

Implémentation

Pour implémenter ces interactions, nous avons utilisé la Kinect Xbox 360 de Microsoft couplée au logiciel de reconnaissance 3-Gear [32] et également le Leap Motion pour détecter les gestes en bordure du système. La majeure partie de notre système a été codé en Java, l’autre partie en C# pour s’interfacer avec Strip’TIC.

Le logiciel 3-Gear (Robert Wang, Chris Twigg & Kenrick Kin) est intéressant pour plusieurs raisons :

- le capteur (type Kinect for Xbox 360) est placé au dessus des bras et des mains de l’utilisateur, permettant la reconnaissance des gestes de l’utilisateur sur l’ensemble de la surface du tableau de strips sans encombrer l’espace de travail de l’utilisateur ;
- le logiciel permet de détecter la position des mains et des doigts de l’utilisateur dans un monde virtuel en trois dimensions avec un bon degré de fidélité ;
- la reconnaissance se fait avec peu de latence et ne nécessite pas une configuration musclée pour avoir un suivi fluide des mouvements ;
- les API en Java, C++ et C# sont complètes et bien documentées.

Automates de gestion des interactions gestuelles

StripEvent	HandEvent	RadarEvent	TableEvent	AreaEvent
StripPress	HandMove	RadarEnter	TablePress	ArchiveAreaEnter
StripRelease	HandFingerMove	RadarLeave	TableRelease	ArchiveAreaLeave
StripPinPress	HandOpen	RadarPointing-ConfigurationEnter	TablePointing-ConfigurationEnter	
StripPinRelease	HandClose	RadarPointing-ConfigurationLeave	TablePointing-ConfigurationLeave	
StripEnter	HandPinch			
StripLeave	HandRelease			
StripMove				
StripMoveStop				

Figure 8 : automates de bas-niveau

Les gestes ont été implémentés par la mise en place d’automates utilisant la librairie SwingStates [1]. Nous avons fait le choix de modéliser les automates complexes nécessaires pour les interactions “métier”, par une composition d’automates simples de plus bas niveau, correspondant à un ensemble de gestes génériques. Cette structuration en “niveaux” permet de construire aisément de nouvelles interactions et de garantir une modularité et réutilisabilité du code. Les automates de bas niveau traitent cinq catégories d’événements gestuels qui concernent respectivement les mains, les strips, le

	1	2	3	4	5
Naturel; Interaction peau	0	1	2	3	0
Naturel; Interaction punaise	1	1	2	2	0
Charge physique; Interaction radar	1	1	1	0	3
Charge cognitive; Interaction radar	1	0	1	1	3

Figure 10 : Distribution par type d'interaction

	1	2	3	4	5
Précision du système	1	2	5	17	11
Réactivité du système	0	2	11	10	13
Apprentissage des interactions	1	0	0	2	33

Figure 11 : Distribution sur l'ensemble des interactions

Deuxième séance

La discussion menée avec les contrôleurs a permis de confronter les interactions à l'avis de contrôleurs professionnels, qui ont l'habitude de travailler avec les systèmes actuels de contrôle. La principale remarque émise par les contrôleurs est la précision trop variable de la détection qui oblige parfois à répéter un geste plusieurs fois pour qu'il soit reconnu. L'ATC étant une activité critique, ce défaut leur semblait rédhibitoire. Ils ont néanmoins apprécié le concept général des interactions proposées et ont reconnu la pertinence de certaines d'entre elles pour le contrôle aérien. Le contrôle de l'image radar, le déplacement de strips virtuels et le pointage à distance ont été par exemple particulièrement appréciés.

DISCUSSION

Performance

Dans l'état actuel, les dispositifs de capture de gestes présentent un grand nombre de fausses détections, ce qui les rend difficilement utilisables pour de vrais produits industriels. D'un point de vue plus radical, on peut considérer que ces dispositifs ne seront jamais utilisables pour des fonctions critiques à cause des ambiguïtés constantes qui peuvent exister dans la communication par gestes. Si la conception proposée propose cependant des fonctions qui ne sont pas réellement critiques, comme la simulation de séquences ou la manipulation de l'image radar, la faible fiabilité des dispositifs limite pour l'instant leur utilité, car les interactions ralentissent l'utilisateur dans ses activités et le contraignent à s'adapter aux capacités de détection du dispositif. Un grand nombre de feedbacks et de feedforwards permettent cependant à l'utilisateur de savoir si le système a compris son action, cela le rassure dans une moindre mesure mais le fruste aussi quand la détection n'est pas fonctionnelle ou erronée. Des dispositifs plus précis feront peut-être leur apparition, ce qui aura pour conséquence des systèmes qui s'adaptent à l'utilisateur et non l'inverse comme dans notre cas d'étude.

Pertinence

Les interactions gestuelles qui ont émergé du travail de conception participative sont essentiellement des interactions qui correspondent à de nouvelles fonctionnalités du système, à part pour les interactions de

Pan & Zoom et de pointage sur l'image radar qui sont possibles dans Strip'TIC avec le stylo Anoto. En effet, il n'existait pas au préalable dans Strip'TIC de système d'archivage des strips virtuels, ni de possibilité explicite de duplication de strips pour simuler des séquences d'arrivée. La possibilité de punaiser un strip virtuel pour les cas où le strip physique est retiré n'existait que sous une forme plus globale, un mode spécial « dégroupement » où le strip virtuel reste affiché après le retrait d'un strip papier. Notre hypothèse est que, au-delà d'une conception de « gestes virtuels » fondés sur la familiarité comme dans [14], que nous avons utilisée notamment pour la punaise et la peau (gestes actuels de maintien et de pince pour saisir), l'utilisation de technologies permettant l'interaction gestuelle dans l'espace aura réellement ouvert des possibilités d'enrichissement des manipulations de strips virtuels, au-delà notamment du déplacement avec le doigt ou le stylo numérique, faisant écho à la richesse de manipulation des strips physiques que nous avons pu observer dans l'activité de contrôle actuelle.

D'autre part, nous pensons que la pertinence du geste conçu dépend aussi de la forme de l'objet. Ainsi, contrairement à des applications décrites par exemple dans [12], où on propose à l'utilisateur des gestes mid-air complétant des interactions tactiles sur une surface pour manipuler des objets ayant un certain volume (comme une tasse), le cas des strips papier est légèrement différent. En effet, un strip papier est certes un objet physique (qu'on peut pointer en le soulevant comme le montre la figure 4.m), mais c'est tout de même un objet plat, faisant bénéficier à son pendant virtuel d'une certaine continuité morphologique, d'une part, mais permettant sans doute aussi de retrouver une continuité gestuelle, celle de ce qu'on pourrait appeler les « gestes papier » de pinch (saisie) ou de punaise (maintien de la feuille).

PERSPECTIVES

Concernant les perspectives, on peut mentionner des aspects activité, des aspects techniques ou de conception et des aspects recherche. Du point de vue des fonctionnalités, pour les contrôleurs il serait intéressant de généraliser la copie de strips, par exemple pour une duplication à l'usage du responsable de salle (supervision, communication). Nous confirmons également l'utilité d'une observation des gestes existants pour mieux comprendre les caractéristiques et les contraintes de l'espace physique de l'activité. Sur le plan de la conception des gestes, il reste notamment à concevoir des gestes associés à plusieurs strips, qui sont fréquents dans l'activité (Figure 4. c, d, h, l, o), ainsi que des gestes collaboratifs (ex. passer un strip d'un contrôleur à un autre). Sur le plan technique, la combinaison effective de la détection multi-touch avec la détection de gestes doit être envisagée. Pour l'instant, le « touché » de la surface est limité à un événement fourni par l'API 3-Gear fonctionnant avec la Kinect Xbox. La précision n'est pas optimale, puisqu'il y a en moyenne un

écart de quelques millimètres avec le tableau de strips. D'autre part, pour la conception hiérarchique d'automates, il faudrait sans doute envisager une utilisation de [2]. Enfin, sur le plan recherche, nous nous intéressons également à la question de la continuité morphologique et interactive entre d'une part les surfaces (table avec affichage et papier), et d'autre part les gestes 2D et 3D, le concept de « geste papier » constituant une métaphore intermédiaire qu'on peut peut-être généraliser aux interactions gestuelles avec des objets plats. Enfin, certains aspects, tels que l'idée de projection d'une icône sur la main, pourront sans doute être repris pour les activités impliquant des actions collaboratives.

CONCLUSION

Dans cet article, nous présentons une exploration des interactions gestuelles dans un environnement particulier : associé à une activité existante (ATC) fortement physique et gestuelle. Il était important d'envisager une approche cohérente et donc plus incarnée, pour la manipulation du versant virtuel de l'interface, celui qui correspond à un affichage sur une surface, par définition moins tangible (au sens de « grasable » [7]). Les résultats de cette étude nous encouragent à continuer cette exploration pour réduire la distance entre des modalités d'entrée encore assez disjointes.

BIBLIOGRAPHIE

- Appert, C. and Beaudouin-Lafon, M. 2008. SwingStates: adding state machines to Java and the Swing toolkit. *Softw. Pract. Exper.* 38, 11 (September 2008), 1149-1182.
- Blanch, R. Programmer l'interaction avec des machines à états hiérarchiques. In *Proc. of IHM '02*, ACM, 129-136.
- Bowman, D.A., Kruijff, E., LaViola, J.J.J., and Poupyrev, I., 3D User Interfaces: Theory and Practice. 1 ed. 2004: Addison-Wesley/Pearson Education.
- Buisson, M, Bustico, A., Chatty, S., Colin, F.R., Jestin, Y., Maury, S., Mertz, C. and Truillet, P. 2002. Ivy: un bus logiciel au service du développement de prototypes de systèmes interactifs. In *Proc. of IHM '02*, ACM, 223-226.
- Cassell, J. A Framework For Gesture Generation And Interpretation. *Computer Vision in Human-Machine Interaction*, p. 191-215.
- Cohn, G., Morris, D., Patel, S., Tan, D.S. Humantenna: Using the Body as an Antenna for Real-Time Whole-Body Interaction. In *Proc. of CHI 2012 Conference on Human Factors in Computing Systems*, p. 1901-1910.
- Fitzmaurice, G., Ishii, H., Buxton, W. Bricks. Laying the foundations for graspable user interfaces. In *Proc. of CHI'95*, ACM.
- Grossman, T. and Wigdor, D. Going deeper: a taxonomy of 3D on the tabletop. In *Proc. of IEEE Tabletop 2007*. p. 137-144.
- Gupta, S., Morris, D., Patel, S., Tan, D.S. SoundWave: Using the Doppler Effect to Sense Gestures, *CHI 2012 Conference on Human Factors in Computing Systems*, p.1911-1914.
- Hancock, M., Carpendale, S., and Cockburn, A. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In *ACM CHI 2007*, ACM. p. 1147-1156.
- Herndon, K.P., Zeleznik, R.C., Robbins, D.C., Brookshire, C.D., Snibbe, S.S., and van Dam, A. Interactive shadows. In *proceedings of UIST 1992*. ACM.
- Hilliges, O., Izadi, S., Wilson, A.D., Hodges, S., Garcia-Mendoza, A., Butz, A. Interactions in the air: adding further depth to interactive tabletops. In *proc of UIST 2009*. ACM.
- Harrison, C., Tan, D.S., Morris, D. Skinput: Appropriating the Body as an Input Surface. In *Proc. of CHI 2010*, ACM. p. 453-462
- Harrison, C., Xiao, R., Schwarz, J. and Hudson, S.E. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction. In *Proc. of CHI '14*, ACM. p. 2913-2916.
- Hurter, C., Lesbordes, R., Letondal, C., Vinot, J.L. and Conversy, S. Strip'TIC: exploring augmented paper strips for air traffic controllers. In *Proc. of AVI '12*, ACM.
- Karam, M. and schraefel, m c. Investigating User Tolerance for Errors in Vision-enabled Gesture-based Interactions. In *Proc. of Working Conference on Advanced Visual Interfaces (2006)*, ACM, p. 225-232.
- Letondal, C., Hurter, C., Lesbordes, R., Vinot, J.L., and Conversy, S. Flights in my hands: coherence concerns in designing Strip'TIC, a tangible space for air traffic controllers. In *Proc. of CHI '13*. ACM.
- Letondal, C., Hurter, C., Lesbordes, R., Vinot, J.L. Strip'TIC : un espace interactif tangible pour les contrôleurs aériens. *Présentation orale à FITG 2012*.
- Mackay, W.E. Is paper safer? The role of paper flight strips in air traffic control. In *proc. of Interact 1999*. ACM.
- Marquardt, N., Jota, R., Greenberg, S. and Jorge, J.A. The continuous Interaction Space. In *Proc of Interact 2011*. ACM.
- Nehaniv, C.L., Dautenhahn, K., Kubacki, J., Haegele, M., Parlitz, C., and Alami, R.. A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction. *Robot and Human Interactive Communication (August 2005)*, 371-377.
- Pereira, J., Jorge, J., Branco, V., Nunes, F. Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons. *WSCG'2000 Conference Proceedings*.
- Rime, B. and Schiaratura, L. Gesture and Speech. *Fundamentals of Nonverbal Behavior*, Feldman & Rime (1991).
- Savery, C., Hurter, C., Lesbordes, R., Cordeil, Graham, T.C.N. When Paper Meets Multi-touch: A Study of Multi-modal Interactions in Air Traffic Controls. In *Proc. of Interact 2013 ACM*
- Shimada, N., Shirai, Y., Kuno, Y., and Miura, J. Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequality constraints. In *Proc of IEEE AFGR*, p268-273.
- Shoemaker, G., Tang, A., and Booth, K.S. Shadow reaching: a new perspective on interaction for large displays. In *ACM UIST'07*. p. 53-56.
- Shneiderman B., Plaisant C., Cohen M. & Jacobs S. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5th ed. Addison-Wesley, USA, 2009.
- Taylor, S., Keskin, C., Hilliges, O., Izadi, S. and Helmes, J. Type-Hover-Swipe in 96 Bytes: A Motion Sensing Mechanical Keyboard, *ACM CHI Conference on Human Factors in Computing Systems 2014*.
- Vermeulen, J., Luyten, K., van den Hoven, E., and Coninx, K. Crossing the Bridge over Normans Gulf of Execution: Revealing Feedforwards True Identity. In *Proc. of CHI 2013*. p. 1931-1940.
- Wilson, A.D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. Bringing physics to the surface. In *UIST'08*. p. 67-76.
- Zimmerman, T.G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. A Hand Gesture Interface Device. In *Proc. of CHI '87 Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, ACM, 189-192.
- 3Gear System Web site: <http://threegear.com>, 3Gear Systems Inc.

Adoiraccourcix : Sélection de Commandes sur Écrans Tactiles Multi-Points par Identification des Doigts

Alix Goguey¹, Géry Casiez², Thomas Pietrzak², Daniel Vogel³ & Nicolas Roussel¹

¹Inria Lille, ²Université Lille 1, France, ³ Université de Waterloo, Canada
 {alix.goguey, nicolas.roussel}@inria.fr, {gery.casiez, thomas.pietrzak}@lifl.fr,
 dvogel@uwaterloo.ca

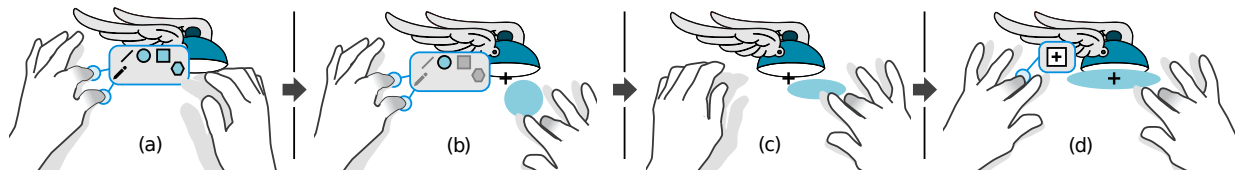


Figure 1: Illustration partielle de la logique d'Adoiraccourcix : (a) un accord de doigts de la main non dominante affiche du *feedforward* précisant les différentes commandes associées aux doigts de la main dominante, pour cet accord. Ici le pouce et l'index affichent les cinq outils de dessin associés aux doigts de la main dominante ; (b) le majeur de la main dominante sélectionne l'outil de création d'ellipses ; (c) le mouvement de la main dominante contrôle les paramètres de la commande, ici la taille et la forme de l'ellipse ; (d) un autre accord de la main non dominante définit une contrainte pour la commande en train d'être exécutée. Ici le majeur de la main non-dominante contraint la création d'ellipse suivant son centre plutôt que son coin supérieur gauche.

RÉSUMÉ

Alors que les raccourcis clavier permettent aux utilisateurs experts des interfaces WIMP d'exécuter rapidement des commandes, il existe aujourd'hui peu d'équivalents pour grands écrans tactiles multi-points. Afin de remédier à ce problème, nous avons conçu Adoiraccourcix, une technique d'interaction tirant parti de l'identification des doigts sur tables interactives, pour la sélection rapide de commandes associée au contrôle continu de paramètres de celles-ci. Après avoir présenté la logique de conception, nous illustrons Adoiraccourcix au travers d'une application de dessin vectoriel. Nous terminons par la présentation d'études préliminaires dans lesquelles nous comparons une mise en œuvre d'Adoiraccourcix à une interface classique. Les premiers résultats suggèrent qu'une fois maîtrisé, Adoiraccourcix permet d'interagir plus efficacement que les interfaces classiques.

Mots Clés

Multi-points ; identification des doigts ; raccourcis ; sélection de commandes ; manipulation directe

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation (e.g. HCI): User interfaces

INTRODUCTION

La sélection de commandes, qui permet à l'utilisateur de réaliser ses tâches en agissant sur des objets virtuels, constitue l'un des aspects essentiels des systèmes interactifs. Elle s'effectue généralement à l'aide de menus et de barres d'outils, mais aussi via des raccourcis clavier qui permettent un accès rapide et facile aux commandes les plus usuelles. Si de tels raccourcis sont communs dans les interfaces WIMP traditionnelles, il existe aujourd'hui peu d'équivalents pour l'interaction sur grands écrans multi-points .

Les boutons physiques étant rares sur les dispositifs tactiles, les concepteurs d'interfaces ont généralement recours à des boutons graphiques, à des techniques prenant en compte le nombre de doigts, les gestes effectués ou encore la temporisation, pour déclencher différentes commandes. Ces approches ne permettent cependant d'accéder qu'à un nombre relativement limité de commandes, elles occupent de la place sur l'écran et peuvent aussi poser des problèmes de performance et de fiabilité.

Pour pallier le manque de techniques d'accès rapide à un grand nombre de commandes sur les écrans tactiles ainsi que le permettent les raccourcis clavier, nous proposons Adoiraccourcix, une technique d'interaction tirant parti de l'identification des doigts en contact avec l'écran qui permet, dans une même et unique interaction, le déclenchement rapide de commandes *et* le contrôle continu des paramètres de celles-ci (Figure 1).

Après une présentation de l'état de l'art sur l'identification des doigts et les systèmes de raccourcis, nous détaillons la logique de conception d'Adoiraccourcix. Nous l'illustrons dans une application de dessin vectoriel, puis nous présentons une expérience utilisateurs préliminaire com-

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670446>

parant une interface traditionnelle à base de boutons graphiques à Adoiraccourcix.

ÉTAT DE L'ART

Nous commençons par présenter les techniques d'identification des doigts en contact avec une surface tactile et les techniques d'interaction qui utilisent cette information. Nous discutons ensuite des avantages de l'interaction bi-manuelle et de la manipulation directe avant de nous concentrer sur les raccourcis clavier et les différentes techniques de sélection de commandes.

Identification des doigts

Les surfaces tactiles utilisant les technologies capacitives et résistives peuvent non seulement fournir la position des contacts mais aussi la pseudo-pression [7] (surface en contact) et l'orientation [25]. Celles qui utilisent une technologie optique, peuvent aussi fournir des informations sur la forme des points de contact [27] ou sur l'utilisateur, la main ou la partie de la main en contact, en ayant recours à des marqueurs [22]. La plupart des techniques d'identification des doigts utilisent la vision par ordinateur.

Les travaux explorant l'identification des doigts ont principalement utilisé cette information pour associer différentes commandes à différents doigts. Sugiura *et al.* utilisent un scanner d'empreintes digitales pour identifier l'index, le majeur, l'annulaire et auriculaire. Ils utilisent ces doigts pour contrôler un lecteur de musique et ajouter des favoris dans un navigateur Internet [23]. En plaçant un électromyogramme sur l'avant-bras, Benko *et al.* identifient chaque doigt en contact avec une précision de 90% [4]. Dans une application de dessin, ils associent une couleur à chaque doigt. Plutôt que de simplement considérer l'utilisation individuelle des doigts, d'autres travaux se sont intéressés à la réalisation de plusieurs contacts simultanément, également appelés accords dans ce qui suit.

Sur des écrans multi-points, Kin-Chung Au *et al.* identifient les doigts en analysant les angles formés par la position des contacts et leur barycentre [1]. Pour correctement identifier les doigts, il est nécessaire de placer les doigts au préalable "naturellement courbés" sur l'écran. Dès lors que cinq contacts simultanés sont détectés et les doigts identifiés, un *Palm menu* composé de boutons associés à chaque doigt est alors affiché en dessous du contact correspondant au doigt et l'utilisateur peut ensuite presser le bouton associé à la commande désirée. L'utilisateur peut exécuter une commande à l'aide d'un doigt ou d'un accord. Ce menu a montré de meilleures performances qu'un menu contextuel et une barre d'outils.

Au lieu d'identifier les doigts, *Arpège* utilise des positions prédéfinies de doigts pour reconnaître des accords [8]. Les accords sont formés progressivement à l'aide de *feedforward*¹, ce qui aide à sélectionner la commande voulue. Les noms des commandes possibles sont affichés près des

¹Dans cet article, le terme *feedback* fait référence à l'information fournie par le système pendant ou après une action de l'utilisateur. Le terme *feedforward* fait référence à l'information fournie avant que l'action ne se déroule. Ces termes anglais sont utilisés à la place de "retour d'information" et "retour d'information par anticipation" pour faciliter la lecture.

doigts et sont mises à jour à chaque changement d'accord. Les commandes sont exécutées quand tous les doigts quittent la surface. Les auteurs ont aussi montré que les utilisateurs sont capables d'exécuter facilement jusqu'à 480 accords. Lepinski *et al.* peuvent reconnaître précisément 14 accords sur une table interactive. Ils ont utilisé ces accords pour créer un *marking menu* tactile [18]. Les utilisateurs effectuent un accord puis déplacent leurs doigts dans une direction pour exécuter une commande.

Au lieu de simplement déclencher des commandes, les accords peuvent être utilisés pour manipuler des outils. Malik *et al.* utilisent deux caméras en hauteur pour identifier les doigts touchant un fond noir [21]. Chaque doigt ou accord peut être associé à une commande et en contrôler les paramètres. Dans leur exemple ils utilisent ce système pour interagir à distance avec un grand écran, par exemple pour zoomer et contrôler le niveau de grossissement. En utilisant un gant équipé de marqueurs, Marquardt *et al.* identifient le bout des doigts, les jointures et d'autres parties de la main [22]. Les utilisateurs peuvent associer un outil ou une commande à chaque marqueur. Ils peuvent ainsi dessiner avec le bout de leur index et avoir un aperçu de l'outil associé en posant la jointure de leur index.

Holz et Baudisch identifient les utilisateurs à l'aide d'un système reconnaissant les empreintes digitales des doigts en contact avec une table interactive [13]. En considérant que chaque doigt a une empreinte digitale unique, cette technique pourrait être utilisée pour identifier les doigts.

En résumé, différentes approches ont été proposées pour identifier les doigts posés sur une surface interactive et déclencher différentes commandes suivant les configurations détectées. Certaines approches pourraient être suffisamment fiables et pratiques dans un avenir proche. Les techniques d'interaction jusqu'ici proposées se sont concentrées sur l'utilisation de doigts seuls ou d'accords pour déclencher des commandes ou manipuler des outils. La découverte des fonctions associées aux doigts et accords est parfois facilitée par du *feedforward*. Aucun de ces travaux de la littérature n'a cependant proposé d'utiliser le *feedforward* dans des techniques d'interaction qui combinent à la fois la sélection de commande et la manipulation directe.

Interaction bi-manuelle et manipulation directe

Identifier les doigts et les mains en contact avec une grande surface permet de bénéficier de la richesse de l'interaction bi-manuelle. Bier *et al.* proposent par exemple le concept de *Toolglass* où la main non dominante manipule une palette transparente de concert avec la main dominante qui contrôle le curseur de la souris [5]. Cliquer avec la souris sur un outil applique la commande correspondante (e.g. une couleur) à l'objet d'intérêt. Dans une expérience comparant les *Toolglasses* à d'autres techniques uni-manuelles Guimbretière *et al.* montrent que la force des *Toolglasses* provient de la combinaison de la sélection de commande et de la manipulation directe. En effet, l'utilisateur peut déclencher une commande et enchaîner avec de la manipulation directe dans un même geste [10]. Vogel et Casiez approfondissent ce concept avec Conté, un crayon digital qui produit différentes actions en fonction

de la face, de l'arrête ou du sommet qui est en contact avec la surface [24]. Le contact du crayon avec la surface sélectionne l'outil et la manipulation directe commence instantanément. Dans *Manual Deskterity*, Hinckley *et al.* explorent la coordination bi-manuelle entre stylet et doigt sur surface tactile où la main non dominante définit le contexte et les outils utilisables avec la main dominante [12]. Par exemple, en fonction de la façon dont un objet est tenu avec les contacts de la main non dominante et du mouvement relatif entre l'objet et le stylet, un trait de stylet peut couper à la manière d'un couteau, tirer un trait le long d'une arête ou créer une copie de l'objet. Ainsi changer d'outil se fait en changeant la position de la main non dominante et non en changeant explicitement d'outil avec la main dominante.

Dans une étude comparative entre palettes flottantes, *marking menus* et *toolglasses*, Mackay montre qu'aucune technique n'est nettement supérieure et plaide pour l'intégration de ces différentes techniques [19]. L'identification des doigts est un moyen élégant de fusionner la sélection de commande et la manipulation directe en fonction de quels doigts touchent quels objets. De plus, cela permet de changer explicitement d'outil en fonction des doigts utilisés.

Raccourcis clavier et sélection de commandes

Les raccourcis claviers donnent aux utilisateurs un accès rapide aux commandes en pressant une ou plusieurs touches simultanément. Les touches de fonction sont ainsi utilisées dans beaucoup d'applications et de systèmes d'exploitation pour déclencher les commandes les plus fréquentes (*e.g.* F1 pour afficher le menu d'aide). Les combinaisons de touches utilisent des modificateurs (*e.g.* Ctrl, Shift) avec d'autres touches pour différencier la sélection de commande de l'entrée de texte (*e.g.* Ctrl+C pour copier un objet). Des modificateurs sont aussi utilisés pour contraindre des actions effectuées à la souris. Par exemple, presser un modificateur dans une application de dessin aime un objet sur une grille ou contraint le rapport hauteur/largeur.

Les principaux problèmes des raccourcis clavier sont leur difficulté à être découverts et retenus. Les utilisateurs ont tendance à ne pas y faire attention, ignorent les indications dans les menus et ne les retiennent pas quand ils ne les utilisent pas fréquemment [17]. Plusieurs techniques permettent d'augmenter leur utilisation, avec un retour visuel [20], auditif [9] ou haptique [3].

La manière la plus courante de déclencher des commandes sur un écran multi-points est d'utiliser les boutons de l'interface graphique. Ces boutons prennent de la place sur l'écran et réduisent la fluidité de l'interaction en obligeant l'utilisateur à effectuer des aller-retours entre les boutons et les objets d'intérêt. Ces deux problèmes peuvent être résolus grâce à l'utilisation de menus contextuels qui sont déclenchés par une action spécifique de l'utilisateur. L'appui long, consistant à maintenir durant un certain temps un ou plusieurs doigts immobiles, est couramment utilisé, par exemple. Il faut cependant que l'utilisateur sache qu'une telle action est possible et le temps d'exécution d'une commande est allongé. Le nombre de doigts en

contact avec la surface interactive est aussi souvent utilisé pour sélectionner la commande, et éventuellement contrôler ensuite ses paramètres. *RST (Rotate-Scale-Translate)*² est un bon exemple d'intégration commande/geste à deux doigts. Le nombre de doigts peut aussi permettre de sélectionner des éléments dans un menu [2]. Une telle méthode permet d'associer jusqu'à 25 commandes, ce qui suffit pour la plupart des applications. Cependant, cette technique rend difficile la découverte des associations, introduit de l'occultation due au nombre de doigts en contact et rend imprévisible l'endroit où sera exécutée la commande. La combinaison de ces techniques avec des gestes permet d'étendre le vocabulaire. Cependant, les utilisateurs doivent d'abord savoir que de tels gestes existent, les gestes sont parfois difficiles à exécuter, leur réalisation prend du temps et leur reconnaissance est sujette à erreurs. Les gestes simples tels que ceux de balayage (*flick*) restent très efficaces quand ils intègrent une transition novice/expert fluide, comme c'est le cas pour les *marking menus* [16].

D'autres informations de contact ont été utilisées pour exécuter des commandes comme la pression exercée, qui est souvent déduite de l'aire de la surface en contact et peut donc être biaisée par d'autres paramètres, comme le nombre de doigts [6]. *Rock & Rails* utilise la forme du contact pour sélectionner des outils [27]. *Tapsense* est un autre exemple qui différencie la pointe, la pulpe, l'ongle et la jointure d'un doigt grâce au son que celui-ci produit lors du contact avec une surface [11]. Cette technique n'occulte pas l'affichage mais elle ne permet pas de distinguer différents doigts.

Dans la section suivante, nous présentons Adoiraccourcix, une technique d'interaction qui utilise l'identification des doigts comme paramètre orthogonal à ceux déjà existants (interactions bi-manuelle, gestes, appuis longs, nombre de doigts) pour la sélection de commandes sur tables interactives.

ADOIRACCOURCIX

Nous détaillons ici la logique de conception d'Adoiraccourcix, créée pour favoriser l'intégration de la sélection de commandes et du contrôle des paramètres associés, en utilisant l'identification des doigts. Dans ce qui suit, nous rappelons qu'un *accord* est considéré comme étant un contact simultané de deux doigts ou plus sur une surface interactive. Une *configuration de contacts*, ou simplement *configuration*, correspond à un contact simple ou à un accord.

Une commande par doigt

L'application la plus évidente de l'identification des doigts est de faire correspondre une commande à chaque doigt [23, 26, 4]. Dans une application de dessin, différents doigts peuvent, par exemple, être utilisés pour créer des lignes, des rectangles ou des cercles en glissant le doigt correspondant sur la surface. D'autres doigts peuvent aussi être utilisés pour déplacer, supprimer ou dupliquer des formes en les touchant. Une telle fusion de la sélection de commande et de ses paramètres (*e.g.* position du doigt pour coller) réduit indéniablement le nombre

²Rotation, changement d'échelle et translation

d'actions nécessaires à la réalisation d'une tâche, comparé à l'utilisation de *widgets* qui peuvent être éloignés de la zone d'intérêt. Cependant cette méthode ne permet d'accéder qu'à 10 commandes.

Utilisation des accords

Avec l'identification des doigts, tout accord de k doigts peut théoriquement se faire de $C(10,k)$ façons. Cela signifie aussi que $C(10,k)$ interactions à k doigts peuvent être sélectionnées en un geste simple. La technique *RST* est généralement effectuée à l'aide du pouce et de l'index de la main dominante, ou des deux index. Avec l'identification des doigts, *RST* pourrait être associée à ces deux accords et les $C(10,2) - 2 = 43$ accords restants pourraient être utilisés pour exécuter d'autres commandes.

Le nombre total d'accords disponibles avec 10 doigts est $\sum_{k=2}^{10} C(10,k) = 1013$. La première ligne de la Table 1 montre la distribution de ce nombre en fonction de k . Cette limite est cependant purement théorique. Le nombre de commandes pouvant être confortablement réalisées à l'aide d'accords est vraisemblablement plus petit [8]. Une étude précise de l'effet complexe des contraintes biomécaniques, des capacités motrices et de mémorisation dépasse le cadre de cet article. Cependant, en considérant la valeur de cette borne supérieure, nous pouvons anticiper des difficultés de mémorisation des couples accords/commandes. Il est donc nécessaire de fournir un moyen de les organiser.

k	1	2	3	4	5	6	7	8	9	10	Total
$C(10,k)$	10	45	120	210	252	210	120	45	10	1	1023
$C(5,k)$	5	10	10	5	1						31
$C(4,k)$	4	6	4	1							15
$C(3,k)$	3	3	1								7
$C(2,k)$	2	1									3

Table 1: Nombre de k -combinaisons de n doigts pour différents valeurs de n .

Approche bi-manuelle inspirée par les raccourcis clavier

Une solution simplifiant la mémorisation et l'exécution d'accords est de réduire le nombre de doigts impliqués et de fournir des moyens de construire les accords de façon incrémentale, en groupant les commandes similaires.

Les *raccourcis clavier* sont en général exécutés à l'aide d'un nombre limité de modificateurs (*e.g.* Ctrl, Shift) et d'une touche symbolique. Les modificateurs ne déclenchent pas de commande mais définissent un mode dans lequel les touches symboliques peuvent être associées à une commande. Ils sont placés principalement sur le bord gauche des claviers et sont de ce fait principalement utilisés par la main non dominante pour les droitiers. L'ordre et le moment suivant lesquels ces modificateurs sont pressés n'a pas d'importance, ce qui facilite leur utilisation.

À l'instar des *raccourcis clavier*, nous proposons d'utiliser les doigts de la main non dominante comme des *modificateurs* et ceux de la main dominante comme *déclencheurs* de commandes. En considérant les ensembles de

configurations S_M de la main non dominante et S_D de la main dominante, le nombre de commandes pouvant être exécutées avec cette approche est :

$$(|S_M| + 1) \times |S_D|$$

avec ou sans nombre de
modificateurs déclencheurs

La partie basse de la Table 1 montre le nombre de k -combinaisons pour 5 à 2 doigts. Un total de 31 accords peuvent théoriquement être effectués avec les cinq doigts d'une main, mais ce nombre chute à 15, 7 et 3 en utilisant seulement 4, 3 ou 2 doigts.

Le nombre total de commandes théoriquement accessibles avec cette approche est $(31 + 1) \times 31 = 992$. Le choix de S_M et S_D est donc crucial pour réduire et organiser l'espace d'interaction. Un grand nombre de modificateurs et un petit nombre de déclencheurs, l'inverse ou un compromis peuvent être préférés suivant le contexte. Pour chaque ensemble, des combinaisons spécifiques de doigts, toutes les combinaisons de certains doigts ou encore une sélection ad-hoc de configurations peuvent être choisies. Par exemple, l'utilisation des combinaisons de 3 ou 4 doigts de la main non dominante et de doigts seuls de la main dominante permet la sélection de $((4 + 6 + 4) + 1) \times 5 = 75$ commandes, ce qui semble suffisant pour la plupart des applications³.

Comme pour les raccourcis clavier, nous anticipons l'émergence de conventions dans les couples commandes/accords bi-manuels pour les opérations communes (*e.g.* copier, couper et coller). Cependant pour la plupart des opérations, S_M and S_D peuvent être spécifiques aux applications. Ainsi, nous devons aussi fournir un moyen pour aider les utilisateurs à comprendre et apprendre l'organisation choisie.

Intégration du feedback et du feedforward

S_M et S_D définissent deux vocabulaires : l'un définissant le mode, l'autre précisant les déclencheurs. Pour chaque mode, à la manière des raccourcis clavier, chaque déclencheur appelle, ou non, une commande. Comme l'utilisateur définit le mode explicitement et de manière incrémentale, il est conscient du changement de mode. Il doit aussi être conscient des conséquences que le mode courant aura sur les futures actions de la main dominante. À cet effet, nous proposons une solution qui intègre *feedback* et *feedforward* : *Lilanotix*.

Lilanotix est affichée à chaque fois qu'un utilisateur effectue une configuration de S_M avec sa main non dominante, avant d'utiliser sa main dominante. Elle disparaît automatiquement quand les doigts de la main non dominante quittent la surface ou que la commande est sélectionnée avec les doigts de la main dominante. Le but de *Lilanotix* est d'indiquer les commandes du mode courant associées aux doigts de la main dominante. La Figure 1 illustre le concept. Dans cet exemple, un accord à deux doigts de la main gauche définit un mode donnant accès à différents outils de dessin associés au pouce, index, majeur, annulaire et auriculaire de la main droite. À noter

³Malacria *et al.* [20] ont compté de 15 à 112 raccourcis clavier sur 30 applications populaires sur Mac ($\mu = 55.31$, $\sigma = 20.53$)

qu'idéalement, *Lilanotix* doit non seulement représenter clairement les commandes disponibles mais aussi à quels doigts de S_D elles sont associées. Le but de *Lilanotix* est d'afficher du *feedforward* et du *feedback*. Les informations affichées ne correspondent en aucun cas à un menu avec des boutons destinés à être pressés. Pour déclencher une commande, l'utilisateur doit exécuter la configuration correspondante avec sa main dominante.

La nature des illustrations de *Lilanotix* varie en fonction des commandes disponibles et de la complexité du vocabulaire de S_D . Par exemple, les icônes déjà présentes dans les barres d'outils peuvent être réutilisées de préférence, par souci de consistance. Pour le vocabulaire de S_D , il est aussi possible de réutiliser des représentations de la littérature, e.g. [18, 8], ou d'en concevoir de nouveaux.

Contrôle de paramètres

Dès que l'utilisateur a exécuté une commande, il peut en contrôler les paramètres en déplaçant ses doigts. Dans le cas le plus simple, les doigts de la main dominante qui ont déclenché la commande sont utilisés pour ce contrôle, procurant ainsi le bénéfice de l'intégration de la sélection de commande et de la manipulation directe [10].

Les commandes associées à un accord peuvent utiliser la trajectoire d'un seul des doigts composant l'accord, le mouvement relatif entre les doigts ou le mouvement global. Une commande déclenchée par le pouce et l'index peut, par exemple, utiliser la distance entre les contacts pour ajuster un paramètre et le barycentre pour en ajuster deux autres. De nouveaux doigts de la main dominante peuvent aussi être pris en compte pour fournir des contrôles additionnels. A noter que la représentation visuelle des commandes dans *Lilanotix* doit être suffisamment explicite pour que les exemples précédents soient compréhensibles.

De nouveaux doigts de la main non dominante peuvent aussi participer au contrôle d'une commande initiée par la main dominante, si *Lilanotix* n'est pas utilisée. Nous avons choisi d'utiliser ces doigts pour effectuer un contrôle particulier. Dans certaines applications, les modificateurs sont utilisés pour contraindre l'interaction effectuée à la souris, par exemple pour aimanter un point sur une grille, aligner horizontalement ou verticalement des objets, forcer le rapport hauteur/largeur, etc. De façon similaire, dès qu'une commande est en cours d'exécution, la main non dominante peut être utilisée pour contraindre l'interaction courante. Les contraintes sont définies par un troisième ensemble de configurations, S_C . Lorsque le nombre de contraintes applicables à une commande est faible, il est possible de les associer à des doigts individuels, de manière à pouvoir facilement combiner ces contraintes en utilisant plusieurs doigts. Un *feedback* spécifique peut apparaître près du ou des doigts concernés par la ou les contraintes (Figure 1, image de droite).

Transition novice-expert

Adoiraccourcix emprunte beaucoup aux raccourcis clavier. Nous émettons l'hypothèse que les utilisateurs familiers avec les raccourcis clavier seront capables de comprendre les principes de base de notre technique.

Le *feedforward* permet d'explorer les accords de la main non-dominante et de connaître les commandes accessibles aux doigts de la main dominante, sans crainte de déclencher une commande par mégarde. Par ailleurs, comme pour les *marking menus*, ce *feedforward* peut être affiché après un certain délai afin de réduire l'occultation de l'écran. Enfin il est envisageable d'offrir à l'utilisateur la possibilité d'afficher explicitement une aide en appuyant par exemple sur un bouton ou en mettant les dix doigts sur la surface. Une fois dans ce mode, l'utilisateur pourrait visualiser et explorer les différentes configurations disponibles avec les commandes associées.

PROTOTYPE D'IDENTIFICATION DES DOIGTS

L'identification fiable des doigts sur une surface multi-points reste un problème technique difficile. L'utilisation d'empreintes digitales est prometteuse [13], mais n'est pas encore opérationnelle pour de grandes surfaces. La vision par ordinateur n'est pas robuste [15, 21], à moins d'utiliser un système de marqueurs attachés sur une paire de gants [22]. Nous présentons ci-dessous le prototype que nous avons mis en place pour obtenir une identification robuste des doigts et ainsi mettre en œuvre Adoiraccourcix.

Notre prototype utilise des *GameTraks* qui sont des contrôleurs de jeu servant à l'origine à simuler des *swings* de club de golf. Cinq *GameTraks* nous permettent de repérer les positions des extrémités de dix doigts dans un espace en 3 dimensions en y attachant les bouts de chacun des fils.

Nous avons écrit une variante d'un serveur TUIO⁴ en C++ et utilisant la librairie *libgametrak*⁵. Cette implémentation établit la correspondance entre la position 3D des doigts et la position 2D des contacts sur un écran 32" 3MTM C3266PW multi-touch surface. À chaque doigt est associé une homographie pour établir cette correspondance. Ces homographies sont calculées dans une phase préalable de calibrage où des points 3D sont associés à une position 2D connue. Une fois le système mis en place, l'identification d'un doigt en contact se fait par la recherche du point 3D le plus proche. Le système génère ensuite un événement TUIO comprenant l'identité du doigt et de la main. Tout le système, ainsi que l'application de dessin vectoriel décrite par la suite, fonctionnent sur un MacBook Pro doté d'un processeur 2.5GHz Intel Core i5 et de 8Go de RAM.

L'architecture de notre système est modulaire, ce qui nous permet d'utiliser d'autres prototypes. Par exemple, nous avons aussi implémenté une identification des doigts avec un *Leap Motion*⁶ fixé au dessus de notre surface tactile. Le *Leap Motion* est capable de repérer l'extrémité des doigts en 3D à 100Hz. La phase de calibrage est similaire à celle précédemment décrite, mais puisque toutes les positions 3D sont données dans le même référentiel, une seule homographie est nécessaire. Cependant, comme la plupart des systèmes basés sur de la vision par ordinateur à un seul point de vue, le *Leap Motion* est sensible aux occultations, ce qui pénalise la fiabilité, même si l'utilisateur

⁴ <http://www.tuio.org/>

⁵ <https://code.google.com/p/libgametrak/>

⁶ <https://www.leapmotion.com>

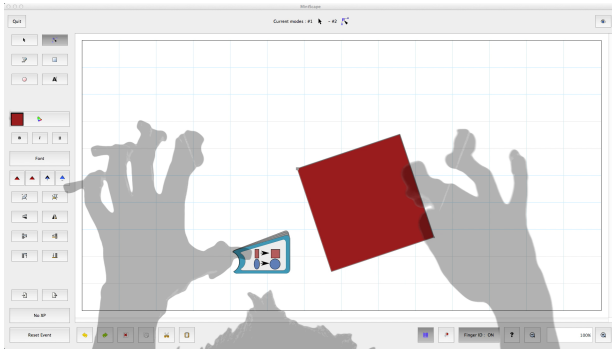


Figure 2: Notre application de dessin vectoriel intègre une GUI, mais elle peut être entièrement contrôlée grâce aux interactions Adoiraccourcix (les mains de l'utilisateur sont représentées sous la forme d'ombres semi-transparentes).

n'est plus attaché. Pour ces raisons, la démonstration suivante et les évaluations préliminaires utilisent la version à base de *GameTraks* qui est plus fiable.

APPLICATION DE DÉMONSTRATION

Afin d'expérimenter Adoiraccourcix, nous avons développé une application de dessin vectoriel dans l'esprit d'Adobe Illustrator⁷ et InkScape⁸. Ce type d'applications a d'abord l'intérêt d'être connu par beaucoup d'utilisateurs et ensuite l'avantage de posséder de nombreuses commandes afin de créer, éditer et manipuler du contenu, ce qui est intéressant pour tester notre technique.

L'application possède six modes et 26 commandes (Figure 2). Les six modes sont : un mode de *sélection* pour la manipulation d'objets (un doigt pour la translation, deux doigts pour le *RST*), un mode d'*édition* pour modifier certains attributs des objets tels que les points de contrôles, et quatre modes de *dessin* pour créer des rectangles, ellipses, courbes et du texte. Afin de comparer les interactions Adoiraccourcix avec une GUI classique, notre application possède des barres d'outils qui permettent l'accès à tous les modes et commandes. En mode expert ces barres d'outils peuvent être cachées afin d'augmenter l'espace de travail disponible à l'écran.

L'application a pour but d'illustrer Adoiraccourcix mais elle met également en œuvre des techniques d'interaction analogues à d'autres techniques de l'état de l'art [22] pour lesquelles aucun mécanisme de *feedforward* n'est disponible. Il s'agit des commandes associées à un seul doigt et des accords uni-manuels.

Par la suite, nous décrirons quels doigts sont utilisés pour accéder aux commandes et aux modes en utilisant la convention suivante : aux noms classiques des doigts nous ajouterons le préfixe 'd-' pour la main dominante et 'nd-' pour la main non dominante. Par exemple, d-index désigne l'index de la main dominante alors que nd-pouce désigne le pouce de la main non dominante.

Commandes associées à un seul doigt

Sélection, édition et création

⁷<http://www.adobe.com/products/illustrator.html>

⁸<http://inkscape.org/>

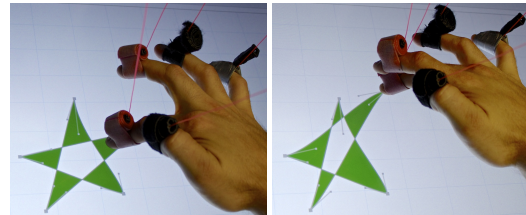


Figure 3: Changement de mode instantané : l'utilisateur déplace une forme avec son d-index et édite ses points de contrôles avec son d-majeur.

Par défaut, l'application est en mode sélection et le d-index est associé à ce mode (Figure 3 gauche) alors que le d-majeur est associé à l'édition de formes et l'ajustement des attributs des objets (e.g. points, Figure 3 droite). Quand l'application passe en mode création (i.e. en mode rectangle), le d-index est alors associé à la création de formes et le d-majeur est utilisé pour sélectionner et déplacer des objets. L'utilisation de l'index et du majeur (plutôt que l'auriculaire par exemple) nous semble adapté à ce type de tâches qui demandent un certain degré de précision dans l'interaction.

Annuler/refaire

Annuler et *refaire* sont des commandes importantes dans les applications de dessin vectoriel. Nous les avons associées au d-pouce combiné avec un geste : d-pouce balayage gauche pour *annuler* et d-pouce balayage droit pour *refaire*.

Accords uni-manuels

Rotation, échelle et translation

Nous avons observé de manière informelle que dans beaucoup d'applications les utilisateurs se servent de la combinaison pouce, index pour les opérations de *RST*. Nous avons donc associé cette combinaison aux manipulations d'objets et au grossissement de vue.

Panoramique

La fonction panoramique permet de naviguer dans un document trop grand pour s'afficher dans son intégralité sur l'écran. De façon similaire à l'utilisation de deux doigts sur un pavé tactile pour faire défiler un document, nous associons aux d-index et d-majeur le contrôle relatif du panoramique. Ce type de contrôle est adapté aux déplacements de faibles amplitudes mais peut demander plusieurs débrayages pour des amplitudes plus importantes. Dans de telles situations, l'utilisation d'un contrôle absolu permet d'atteindre plus rapidement une région d'intérêt sans avoir recours au grossissement. Nous avons choisi d'associer aux d-majeur et d-annulaire le contrôle absolu du panoramique, où la vue est centrée sur le point du document correspondant à la position absolue des doigts sur l'écran.

Loupe et tiroir

La combinaison d-pouce et d-index contrôle le niveau de grossissement de la vue sur le document. Cependant il est parfois préférable de n'agrandir qu'une petite partie du document pour y ajuster des détails, sans perdre le contexte général. Pour cela, nous avons implémenté une loupe (Figure 4), comme détaillé par Käser *et al.* [14] et mis en œuvre dans certaines applications commerciales

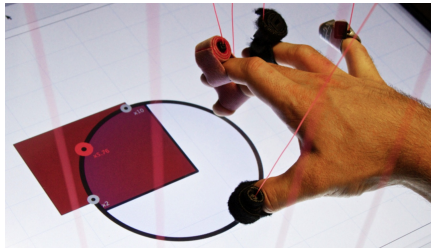


Figure 4: Loupe : grossissement d'une partie de la feuille de dessin. L'utilisateur peut ajuster le niveau de grossissement à l'aide du curseur et déplacer la loupe en glissant les doigts depuis le bord.

comme *zoom* dans *Paper*⁹. Cette loupe est associée à la combinaison d-pouce et d-majeur. Les utilisateurs peuvent déplacer et ajuster la taille de la loupe en déplaçant ces doigts. Un curseur situé sur le bord de la loupe et ajustable par n'importe quel doigt de S_D contrôle le niveau de zoom. De plus tous les doigts de S_D peuvent déplacer la loupe en la glissant depuis son bord. Une fois apparue, la loupe reste accessible jusqu'à ce que l'utilisateur la fasse disparaître en réduisant sa taille ou en effectuant une double tape sur celle-ci avec d-pouce et d-majeur.

Nous avons aussi développé un tiroir destiné à stocker des objets graphiques qui pourront par la suite être restitués ou copiés. L'utilisateur peut copier un objet ou un de ses attributs comme la couleur, le type de contour ou le formatage d'un texte. Selon le même principe que la loupe, le tiroir est appelé avec la combinaison d-pouce et d-auriculaire (Figure 5). Les objets peuvent ensuite être glissés dans le tiroir. L'objet est soit déplacé ou copié si le d-annulaire est en contact avec la surface lors de la fin du glissé. De la même manière, déplacer ou dupliquer les objets depuis le tiroir s'effectue en glissant les objets du tiroir vers la feuille de dessin. Pour copier un attribut, l'utilisateur doit changer le mode courant associé au tiroir en maintenant le d-pouce sur la surface et en réalisant une tape sur le tiroir avec le d-auriculaire. Les modes défilent de façon circulaire entre : *forme* (décrit ci-dessus), *couleur*, *type de contour*, *formatage de texte*.

Accords bi-manuels

Les accords uni-manuels sont pratiques pour exécuter les commandes les plus fréquemment utilisées. Pour étendre le nombre de commandes disponibles à l'aide d'accords tout en aidant les utilisateurs à ne pas s'y perdre, nous utilisons Adoiraccourcix avec une mise en œuvre de *Lilannotix*.

Avec un total de 29 commandes et modes restants à associer à des accords, nous avons choisi de les rassembler dans 7 groupes comprenant au maximum 5 commandes (Table 2). Selon la Table 1, 3 doigts de la main non dominante sont suffisants pour différencier 7 catégories ($|S_M| = 3 + 3 + 1 = 7$) dans lesquelles chacun des 5 doigts de la main dominante exécute une commande différente (en supposant qu'il n'y a pas d'accords de doigts avec la main dominante). En faisant ce choix, nous pouvons sélectionner $(7 + 1) \times 5 = 40$ commandes, ce qui est suffisant pour notre exemple. Nous utilisons les accords des nd-pouce, nb-index et nd-majeur pour accéder à ces

⁹<http://www.fiftythree.com/paper>

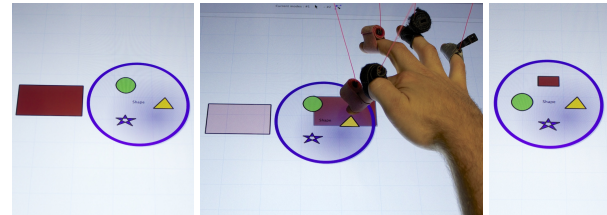


Figure 5: Tiroir : les objets peuvent être glissés sur le tiroir pour y être stockés.

7 catégories. Une fois un de ces accords joué, du *feedforward* apparaît en fondu après 0.3s et montre les commandes accessibles aux doigts de la main dominante (Figure 6). Le *feedforward* est constitué d'un cercle de couleur affiché en dessous de chaque doigt composant l'accord. Ces cercles sont reliés à un rectangle comportant 5 icônes représentant les doigts de la main dominante. Par exemple, l'accord nd-pouce + nd-index affiche le *feedforward* de l'image de droite de la Figure 6, indiquant que le d-pouce correspond à la multi-sélection, d-index à la commande copier, d-majeur à couper, d-annulaire à coller et d-auriculaire à la commande effacer. Quand les doigts de la main dominante sont associés à des gestes, ceux-ci sont représentés au-dessus de l'icône correspondante. Par exemple, les commandes *annuler* et *refaire* sont aussi accessibles via l'accord nd-pouce combiné à un balayage du d-pouce. Les balayages correspondants sont représentés au dessus de l'icône pouce (Figure 6 image de gauche).

Un des accords (nd-index + nd-majeur) est associé à des groupes contextuels. Les commandes disponibles pour la main dominante dépendent de l'objet sélectionné. Si une ou plusieurs formes sont sélectionnées, les commandes disponibles sont : multi-sélection, contour plein, contour pointillé, contour plus épais et contour plus fin. Si un ou plusieurs textes sont sélectionnés, les commandes disponibles sont : multi-sélection, gras, italique, souligné et police. Quand aucun objet n'est sélectionné ou si les deux types d'objets sont sélectionnés aucune commande n'est disponible.

Sélecteur de couleur

Pour la sélection de couleurs nous avons adapté une palette couleurs classique que l'utilisateur appelle avec les nd-pouce et nd-auriculaire (Figure 7). L'utilisateur ajuste la teinte, lumière, saturation et transparence avec n'importe quel doigt de la main dominante, ce qui met à jour la couleur de l'arrière plan de la roue. L'utilisateur peut ensuite appliquer la couleur courante au remplissage d'un objet en le touchant avec le d-index, et à son contour en le touchant avec le d-majeur. Il peut aussi modifier la couleur courante en touchant un objet avec le d-annulaire pour obtenir sa couleur de remplissage et avec le d-auriculaire pour obtenir la couleur de son contour.

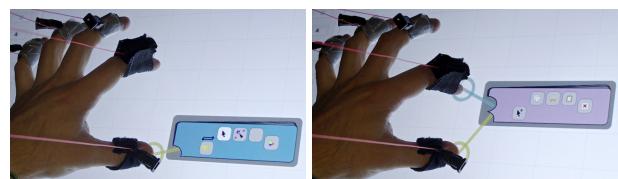


Figure 6: *Lilannotix* : (a) commandes associées aux doigts de la main dominante en association avec nd-pouce ; (b) commandes copier/couper/coller/effacer associées à l'accord nd-pouce et nd-index.

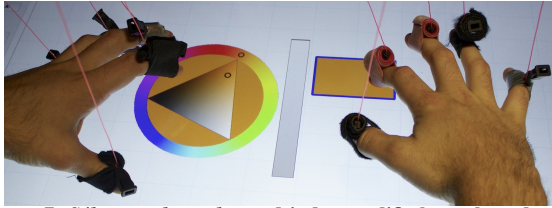


Figure 7: Sélection de couleur : d-index modifie la couleur de remplissage du rectangle, d-majeur peut modifier celle du contour, d-annulaire permet d'obtenir la couleur de remplissage et d-auriculaire celle du contour.

Application de contraintes

Quand l'utilisateur interagit avec les doigts de la main dominante, ceux de la main non dominante peuvent définir des contraintes. Lors de la création ou de l'édition de formes, nous avons associé le nd-pouce pour contraindre le rapport hauteur/largeur à des formes parfaites (e.g. carré vs. rectangle). Lors de la manipulation, le nd-index est utilisé pour désactiver le changement d'échelle de l'objet. Enfin pour tous les modes, le nd-majeur fixe le centre de la forme et le nd-annulaire aimante les points d'intérêts sur la grille. Comme chaque contrainte est associée à un doigt différent, les contraintes peuvent être combinées. Un feedback présenté sous la forme d'une étiquette est affiché près de chaque doigt de la main non dominante en cours d'utilisation (Figure 8). Ces étiquettes sont grisées lorsque la contrainte ne peut pas être appliquée.

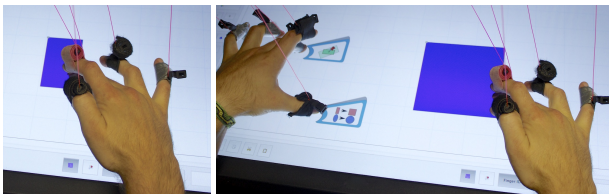


Figure 8: Contraintes de manipulation : elles apparaissent lorsque qu'un doigt de la main non-dominante est en contact avec l'écran alors qu'un ou plusieurs doigts de la main dominante manipulent un objet.

Transition novice-expert

Comme décrit précédemment, nous avons apporté un soin particulier à utiliser du feedforward et du feedback quand cela est utile à la découverte des associations accords/commandes. En plus de cela, nous avons ajouté

Accord	pouce	index	majeur	annulaire	auriculaire
○○○○●		Sélection	Modification		Couleur
○○●○○		Courbe	Rectangle	Cercle	Texte
○○●○○	S+	Devant	Vers l'avant	Vers l'arrière	A l'arrière
○○○●●	S+	Copier	Couper	Coller	Effacer
○○●○○	S+	Grouper	Dégrouper		
○○●○○	S+	← commandes contextuelles →			
○○●●●	S+	Vsym	Hsym		
●○○○○		← outils couleur →			
○○○○○	●	Zoom	Loupe		Tiroir

Table 2: Liste des associations de Lilanotix. L'accord effectué avec la main non dominante est indiqué dans la colonne de gauche où les cercles représentent de gauche à droite les doigts de l'auriculaire au pouce. ● et ○ représentent respectivement des doigts en contact ou pas. Les commandes associées aux doigts de la main dominante sont listées dans les colonnes suivantes où S+ représente la multi-sélection.

des info-bulles pour chaque bouton de la GUI montrant l'équivalent Adoiraccourcix. Ces info-bulles apparaissent à proximité des boutons quand ils sont pressés plus d'une seconde. Enfin, un mode d'aide peut être explicitement appelé par l'utilisateur lorsqu'il presse un bouton d'aide ou lorsqu'il pose ses 10 doigts simultanément sur la surface. Dans ce mode, des info-bulles spécifiques à chaque doigt sont affichées près de ceux-ci (Figure 9). Les info-bulles affichent la liste des icônes des commandes associées à chaque doigt et la liste des doigts composant les combinaisons (le préfixe "Opp." indique des doigts de la main opposée). Si l'utilisateur souhaite explorer une commande, il lui suffit de jouer l'accord et les info-bulles afficheront plus de détails. Pour quitter ce mode, soit l'utilisateur presse le bouton d'aide, soit il effectue une double tape sur la surface avec ses 10 doigts.

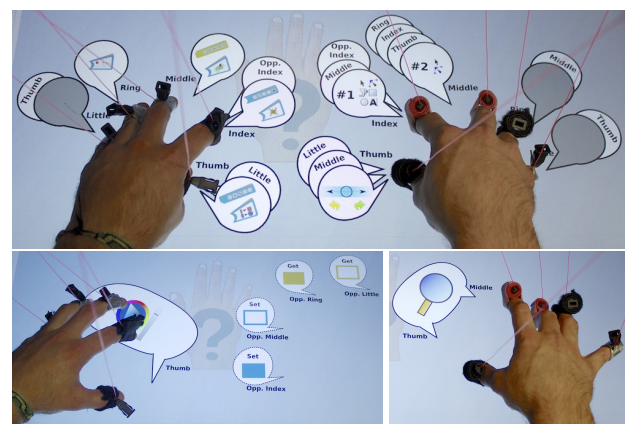


Figure 9: Mode d'aide. Haut : chaque doigt possède des bulles indiquant l'action associée à ce doigt, ou les combinaisons avec les autres doigts. Bas : exemples d'exploration en utilisant certaines combinaisons avec à gauche le sélecteur de couleur et à droite la loupe.

EXPÉRIENCES PRÉLIMINAIRES

Nous avons réalisé deux expériences préliminaires, la première pour estimer dans quelle mesure des utilisateurs novices sont capables de découvrir les différentes fonctionnalités offertes par Adoiraccourcix dans notre application de dessin vectoriel, et la seconde pour mesurer les performances d'Adoiraccourcix en comparaison avec une GUI classique dans le cadre d'une utilisation experte.

Dans la première expérience, 8 volontaires (20-35 ans, 2 femmes, 2 non informaticiens), tous droitiers, ont participé à une session de pensée à voix haute d'une heure environ comportant 2 parties. La première partie était une période de temps libre pendant laquelle un expérimentateur rappelait aux participants le fonctionnement basique des applications de dessin vectoriel, et où les participants découvraient et interagissaient, sans limite de temps, avec l'application, à la fois avec Adoiraccourcix et la GUI. Durant cette phase, quand cela était nécessaire, l'expérimentateur les aidait à comprendre comment utiliser chaque fonctionnalité et leur montrait celles qui n'avaient pas été découvertes. Dans la seconde partie, les participants étaient invités à effectuer quatre tâches simples en utilisant la GUI ou Adoiraccourcix : créer des rectangles, des courbes et des ellipses selon des dimensions données, dupliquer des objets pour créer un ciel étoilé, changer les couleurs de différents objets et dessiner une maison en

partant d'une feuille blanche (les seules indications étaient qu'elle comporte une porte avec une poignée, deux fenêtres et un toit). À la fin, les participants ont répondu à un questionnaire sur chacune des parties dans le but de collecter leurs impressions en répondant à des questions sur une échelle de Likert à 5 points.

Durant la période de temps libre, nous avons collecté les remarques des participants à propos de l'application. Tous les participants ont découvert d'eux-mêmes *Lilano-tix*. Nous avons cependant observé que certains participants ont essayé d'appuyer sur les icônes comme si celles-ci étaient des boutons. Pour remédier à ce problème, l'affordance de l'affichage actuel pourrait être amélioré en estompant par exemple l'aspect bouton des icônes. Pour le tiroir, nous avons dû expliquer le rôle des d-annulaire et d-auriculaire. De même certains participants n'ont pas découvert toutes les interactions uni-manuelles d'eux mêmes à cause du manque de *feedforward*. Quant au questionnaire, les utilisateurs ont confirmé la ressemblance de notre application avec celles de dessin vectoriel qu'ils connaissaient (4.4/5). Ils ont réussi à effectuer les tâches qu'ils souhaitaient (4.5/5). Leurs outils préférés d'Adoiraccourcix (min : 3.4/5, max : 4.7/5, médiane : 4.0/5) étaient le *sélecteur de couleur* (4.7/5), *Lilano-tix* (4.5/5) et le *tiroir* (4.3/5). Le moins aimé était le *contrôle absolu du panoramique* (3.4/5). L'impression sur la facilité d'utilisation est au dessus de la moyenne (min : 3.6/5, max : 4.3/5, médiane : 3.8/5). Les participants nous disaient avoir besoin de temps pour apprendre les différents raccourcis mais tous ont affirmé avoir le sentiment qu'ils amélioreraient leur performance avec plus de pratique. Ils ont aussi fait part d'une plus grande fluidité d'interaction qu'avec une *GUI* (4.0/5).

En considérant que notre interface cible prioritairement des utilisateurs experts, nous n'étions pas surpris que les participants n'aient pas atteint un certain niveau d'habileté en une heure. Pour mesurer le gain de performance des utilisateurs experts, nous avons recruté des utilisateurs pour réaliser de manière répétitive des tâches simples en utilisant Adoiraccourcix et la *GUI* : 1) dupliquer un objet et le déplacer à une position donnée, 2) créer un rectangle dont les sommets opposés doivent être à une position donnée, et 3) éditer une courbe et déplacer un des points à une certaine position. Au début de chaque essai l'application était en mode *sélection*. Chaque tâche comprenait 5 blocs de 10 essais où les conditions (*i.e.* positions cibles) à chaque essai étaient identiques. De plus l'expérimentateur imposait aux participants l'utilisation de la façon la plus rapide de réaliser chaque tâche pour chaque technique (*i.e.* utiliser la main non dominante pour presser les boutons de la *GUI* et la main dominante pour interagir). Les conditions favorisaient une utilisation experte des deux SYSTÈMES. Nous avons mesuré le temps de chaque essai, de la première interaction détectée à la fin de l'essai. Nous avons utilisé un plan expérimental intra-sujets et l'ordre des conditions était balancé parmi les participants (Adoiraccourcix et la *GUI*). 12 participants (23-35 ans, 1 femme, 3 non informaticiens), tous droitiers, ont pris part à l'expérience. Le plan expérimental était : 12 PARTICIPANTS \times 2 SYS-

TÈMES \times 3 TÂCHES \times 5 BLOCS \times 10 RÉPÉTITIONS = 3600 essais.

Comme les temps enregistrés ne suivaient pas une loi normale, les données ont été traitées en utilisant un *Aligned Rank Transform* [28] avant d'y appliquer une ANOVA à mesures répétées. L'analyse des résultats a montré un effet significatif du SYSTÈME ($F_{1,11} = 38.7$, $p < 0.001$) et une interaction significative SYSTÈME \times TÂCHE ($F_{2,22} = 48.8$, $p < 0.001$) sur le temps de réalisation de la tâche. Des analyses post-hoc ont révélé des différences significatives entre les deux systèmes pour la première tâche ($p < 0.001$, *GUI* : 1.87s, Adoiraccourcix : 1.03s) et la troisième ($p < 0.001$, *GUI* : 1.14s, Adoiraccourcix : 0.71s) contrairement à la seconde tâche (*GUI* : 0.91s, Adoiraccourcix : 0.92s). Les différences significatives observées pour les tâches 1 et 3 peuvent être expliquées par le nombre plus élevé d'actions à effectuer dans la condition *GUI* par rapport à la condition Adoiraccourcix (appuyer sur les boutons copier et coller ainsi que de changement de mode). Dans la seconde tâche, avec les deux systèmes, il n'y avait qu'un changement de mode à effectuer. Adoiraccourcix troque l'appui de bouton et autres actions moteurs de la *GUI* contre la mémorisation et l'exécution d'accords.

LIMITATIONS

Bien que le *feedforward* de *Lilano-tix* permette aux utilisateurs de trouver facilement la commande souhaitée, il faut néanmoins savoir quels sont les doigts qui définissent l'ensemble S_M . De même, il n'existe pas de *feedforward* pour les accords uni-manuels. Cependant une fois ces informations connues, le reste des commandes est facile à découvrir. De plus, le mode d'aide et les *info-bulles* favorisent cette découverte.

Notre but n'est pas de donner une association de commandes idéale mais d'illustrer notre logique de conception. Nos choix d'associations sont améliorables. Par exemple l'association de commandes que nous avons proposée suppose que l'utilisateur est droitier. Dans un contexte écologique nous suggérons de permettre à l'utilisateur de configurer sa main dominante, en intégrant par exemple un bouton sur l'interface.

Notre application ne supporte pas la personnalisation des associations accords/commandes. Cependant cette possibilité n'est pas incompatible avec Adoiraccourcix. À la manière des raccourcis clavier, nous suggérons aux développeurs d'applications de permettre aux utilisateurs de configurer l'association des commandes tout en respectant la logique de conception Adoiraccourcix.

CONCLUSION ET TRAVAUX FUTURS

En partant du constat qu'il existe peu d'équivalents des raccourcis clavier sur grands écrans multi-points, nous avons proposé de tirer parti de l'identification des doigts pour favoriser non seulement la sélection rapide de commandes mais aussi le contrôle combiné des paramètres qui leurs sont associés. Inspiré par les raccourcis clavier, nous avons proposé d'associer les doigts de la main non dominante à des modificateurs et ceux de la main dominante à des outils. Cette séparation permet d'introduire *feedforward* et *feedback* pour faciliter la découverte des

accords existants. Nous avons illustré Adoiraccourcix au travers d'une application de dessin vectoriel et les résultats d'études préliminaires suggèrent qu'Adoiraccourcix offre une alternative prometteuse aux interfaces classiques. Pour les travaux futurs nous envisageons d'explorer l'utilisation de l'identification des doigts dans d'autres contextes que les tables interactives. Nous envisageons aussi de développer des prototypes plus pratiques à utiliser de manière à pouvoir intégrer Adoiraccourcix dans des applications existantes et ainsi mesurer l'intérêt de notre approche de manière plus approfondie. Enfin, nous souhaitons mettre en place des expériences vérifiant si nos hypothèses d'ergonomie sont justes et si *Lilanotix* aide à la mémorisation.

REFERENCES

1. Au, O. K.-C., and Tai, C.-L. Multitouch finger registration and its applications. In *Proc. OZCHI'10*, 41–48.
2. Bailly, G., Lecolinet, E., and Guiard, Y. Finger-count & radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces. In *Proc. CHI'10*, 591–594.
3. Bailly, G., Pietrzak, T., Deber, J., and Wigdor, D. J. Métamorphe: augmenting hotkey usage with actuated keys. In *Proc. CHI'13*, 563–572.
4. Benko, H., Saponas, T. S., Morris, D., and Tan, D. Enhancing input on and above the interactive surface with muscle sensing. In *Proc. ITS'09*, 93–100.
5. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. In *Proc. SIGGRAPH'93*, 73–80.
6. Cao, X., Wilson, A., Balakrishnan, R., Hinckley, K., and Hudson, S. Shapetouch: Leveraging contact shape on interactive surfaces. In *Proc. TABLETOP'08*, 129–136.
7. Davidson, P. L., and Han, J. Y. Extending 2d object arrangement with pressure-sensitive layering cues. In *Proc. UIST'08*, 87–90.
8. Ghomi, E., Huot, S., Bau, O., Beaudouin-Lafon, M., and Mackay, W. E. Arpège: Learning multitouch chord gestures vocabularies. In *Proc. ITS'13*, 209–218.
9. Grossman, T., Dragicevic, P., and Balakrishnan, R. Strategies for accelerating on-line learning of hotkeys. In *Proc. CHI'07*, 1591–1600.
10. Guimbretière, F., Martin, A., and Winograd, T. Benefits of merging command selection and direct manipulation. *ACM ToCHI* 12, 3 (Sept. 2005), 460–476.
11. Harrison, C., Schwarz, J., and Hudson, S. E. TapSense: enhancing finger interaction on touch surfaces. In *Proc. UIST'11*, 627–636.
12. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. Pen + touch = new tools. In *Proc. UIST'10*, 27–36.
13. Holz, C., and Baudisch, P. Fiberio: A touchscreen that senses fingerprints. In *Proc. UIST'13*, 41–50.
14. Käser, D. P., Agrawala, M., and Pauly, M. FingerGlass: efficient multiscale interaction on multitouch screens. In *Proc. CHI'11*, 1601–1610.
15. Kung, P., Küser, D., Schroeder, C., DeRose, T., Greenberg, D., and Kin, K. An augmented multi-touch system using hand and finger identification. In *CHI EA'12*, 1431–1432.
16. Kurtenbach, G., and Buxton, W. The limits of expert performance using hierarchic marking menus. In *Proc. CHI'93*, 482–487.
17. Lane, D. M., Napier, A. H., Peres, C. S., and Sándor, A. Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. *IJHCI* 2, 18 (2005), 133–144.
18. Lepinski, G. J., Grossman, T., and Fitzmaurice, G. The design and evaluation of multitouch marking menus. In *Proc. CHI'10*, 2233–2242.
19. Mackay, W. Which interaction technique works when? Floating palettes, marking menus and toolglasses support different task strategies. In *Proc. AVI'02*, 203–208.
20. Malacria, S., Bailly, G., Harrison, J., Cockburn, A., and Gutwin, C. Promoting hotkey use through rehearsal with ExposeHK. In *Proc. CHI'13*, 573–582.
21. Malik, S., Ranjan, A., and Balakrishnan, R. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proc. UIST'05*, 43–52.
22. Marquardt, N., Kiemer, J., Ledo, D., Boring, S., and Greenberg, S. Designing user-, hand-, and handpart-aware tabletop interactions with the TouchID toolkit. In *Proc. ITS'11*, 21–30.
23. Sugiura, A., and Koseki, Y. A user interface using fingerprint recognition: holding commands and data objects on fingers. In *Proc. UIST'98*, 71–79.
24. Vogel, D., and Casiez, G. Conté: multimodal input inspired by an artist's crayon. In *Proc. UIST'11*, 357–366.
25. Wang, F., Cao, X., Ren, X., and Irani, P. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proc. UIST'09*, 23–32.
26. Wang, J., and Canny, J. FingerSense: augmenting expressiveness to physical pushing button by fingertip identification. In *CHI EA'04*, 1267–1270.
27. Wigdor, D., Benko, H., Pella, J., Lombardo, J., and Williams, S. Rock & rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *Proc. CHI'11*, 1581–1590.
28. Wobbrock, J. O., Findlater, L., Gergle, D., and Higgins, J. J. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proc. CHI'11*, 143–146.

CtrlMouse et TouchCtrl : Dupliquer les Délimiteurs de Mode sur la Souris

Thomas Pietrzak
Université Lille 1
thomas.pietrzak@lfl.fr

Sylvain Malacria
University College London
sylvain@malacria.fr

Gilles Bailly
CNRS, Télécom ParisTech
gilles.bailly@telecom-
paristech.fr

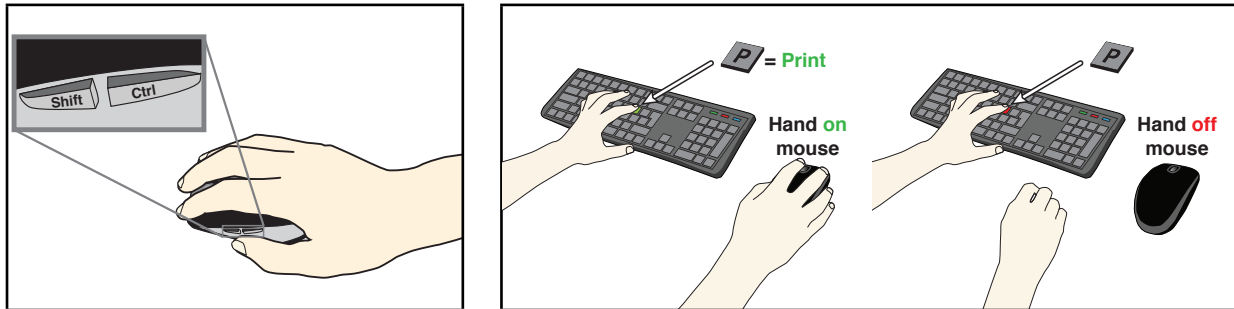


Figure 1: Gauche : CtrlMouse duplique les touches Ctrl et Shift sur la souris pour délimiter l'entrée de texte de la sélection de commandes et réduire le besoin de coordination des doigts de la main gauche. Droite : TouchCtrl déclenche automatiquement la touche Ctrl lorsque l'utilisateur a sa main posée sur la souris.

RESUME

Les touches modificateur du clavier comme Ctrl ou Cmd (⌘) sont utilisées pour délimiter l'entrée de texte de la sélection de commandes (raccourcis claviers). Dans cet article nous étudions l'impact de la localisation de ces délimiteurs sur la performance et la charge musculaire en les dupliquant sur la souris. À cet effet, nous dérivons deux techniques d'interaction : CtrlMouse duplique les touches Ctrl et Shift en les associant aux boutons de la souris sous le pouce ; TouchCtrl déclenche automatiquement la touche Ctrl lorsque la main est posée sur la souris. Deux expériences en laboratoire révèlent 1) que ces techniques sont d'autant plus adoptées par les utilisateurs que la tâche demande du pointage, 2) le coût temporel des modificateurs sur la sélection de commandes est de 0,21s, ce qui correspond à 11,9% du temps d'exécution et 3) le temps d'exécution avec CtrlMouse avec un ou deux modificateurs est similaire. Nous avons également déployé ces techniques pour valider de manière écologique les résultats obtenus lors des évaluations en laboratoire. Enfin, nous présentons différents scénarii applicatifs élaborés à partir de CtrlMouse et TouchCtrl.

Mots Cles

modificateurs, délimiteurs de mode, raccourcis clavier

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation (e.g. HCI): Input devices and strategies.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670447>

INTRODUCTION

Les claviers sont à la fois utilisés pour saisir du texte et pour sélectionner des commandes à l'aide de raccourcis clavier. Le passage entre deux modes se fait à l'aide de *délimiteurs*. Typiquement, les utilisateurs passent en mode commande en pressant des touches *modificateur* telles que Ctrl ou Cmd (⌘). Ces deux modes ont été amplement et indépendamment étudiés ([3, 11, 12, 29, 45]) mais il y a étonnamment peu de travaux sur la *transition* entre ces deux modes. Pourtant, l'action d'appuyer sur une ou plusieurs touches faisant office de délimiteur de mode influence directement la performance de l'utilisateur en termes de vitesse, précision et confort. En effet, appuyer sur une touche ralentit l'interaction. D'après le modèle GOMS [8], appuyer sur ce modificateur compte comme une opération d'appui de touche séparée comptabilisant pour 0,28s par sélection. Bien que cette pénalité puisse être modeste, elle peut s'avérer importante pour des utilisateurs experts sélectionnant plusieurs centaines voire milliers de commandes par jours. D'autre part, exécuter un raccourci clavier peut introduire une charge musculaire inconfortable pour l'utilisateur [37]. En effet, certains raccourcis (ex : Ctrl + Shift) nécessitent une coordination délicate des doigts pour réaliser des accords [37].

Dans ce papier, nous étudions l'impact de la localisation des modificateurs sur la performance et le confort de l'utilisateur. Plus précisément, nous étudions l'utilisation de la souris comme alternative pour délimiter l'entrée de texte de la sélection de commandes. Pour cela, nous proposons CtrlMouse, une technique d'interaction qui duplique les modificateurs Ctrl et Shift sur les boutons du pouce de la souris, (Figure 1-gauche). Les utilisateurs peuvent donc basculer *explicitement* entre les modes de saisie de texte et raccourcis claviers grâce à deux boutons dédiés. Nous présentons également TouchCtrl, une méthode alternative qui déclenche *implicitement* le mode sélection de commandes lorsque la main est posée sur la souris (Figure 1-

droite). Ces deux techniques peuvent être déployées sur un grand nombre d'applications avec une souris ayant des boutons sous le pouce (comme des souris standard Logitech [26]), ou une souris tactile comme l'Apple Magic-Mouse [1] ou la Microsoft TouchMouse [33].

Nous présentons deux évaluations en laboratoire étudiant l'adoption (étude 1) et la performance (étude 2) de Ctrl-Mouse et TouchCtrl. Les résultats 1) montrent que ces techniques sont d'autant plus adoptées par les utilisateurs que la tâche nécessite des opérations de pointage plutôt que de l'entrée de texte ; 2) estiment le coût temporel des modifieurs en fonction de sa *localisation* (souris et clavier), de son mode d'*activation* (implicite ou explicite) et du nombre de modifieurs utilisés.

Nous avons également réalisé une étude qualitative préliminaire sur le terrain pour confirmer de manière écologique les avantages de CtrlMouse et TouchCtrl lors des évaluations en laboratoire. Nous avons déployé nos deux techniques d'interaction auprès de 6 utilisateurs. Les résultats qualitatifs sont partagés parmi les utilisateurs. CtrlMouse est perçu comme un avantage dans le cas des raccourcis à plusieurs modifieurs ou pour les applications graphiques. Inversement, l'implémentation logicielle et matérielle a un impact sur l'utilisabilité.

Enfin nous décrivons 3 extensions tirant parti de Ctrl-Mouse et TouchCtrl : 1) Nous étendons le principe de TouchCtrl au pavé tactile. 2) Nous discutons l'utilisation de nos techniques dans d'autres contextes (sélection d'objets) et 3) avec d'autres techniques d'interaction.

En résumé nos contributions incluent 1) un nouveau regard quantitatif et qualitatif sur les délimiteurs de modes à travers deux évaluations en laboratoire et une évaluation préliminaire sur le terrain ; 2) la conception, la réalisation et l'évaluation de deux techniques d'interaction ; 3) 3 exemples d'applications pour généraliser les concepts introduits par CtrlMouse et TouchCtrl.

ETAT DE L'ART

Ce projet contribue dans le domaine des délimiteurs entre le mode d'entrée de texte et le mode de sélection de commandes. Étant donné que notre approche consiste à modifier le comportement d'un dispositif d'entrée traditionnelle (la souris), nous discutons aussi les travaux qui augmentent les souris et claviers avec de nouvelles capacités.

Délimiteurs et sélecteurs de modes

Les *délimiteurs* ont été étudiés à plusieurs reprises [14, 17, 21, 36, 40, 47, 24, 42, 39] en particulier dans le contexte de l'interaction gestuelle et au stylet. Ils permettent aux systèmes interactifs de déterminer la structure lexicale d'une phrase d'entrée [7] et de distinguer différents modes. La plupart des applications que l'on utilise sur nos ordinateurs sont par défaut en mode de saisie de texte : lorsque l'on presse une touche du clavier, la lettre correspondante est saisie et aucune commande est activée. Plusieurs touches du clavier sont alors utilisées comme délimiteurs. Typiquement les touches *modifieurs* permettent de changer l'action d'une touche à l'aide de pseudo-modes. Cependant d'autres touches spéciales permettent de changer de mode. Ainsi *Shift* (pseudo-mode)

et *Caps* (mode) permettent de basculer entre minuscules et majuscules. La touche *Ctrl* permet de délimiter le mode de saisie de texte et le mode de sélection de commandes. D'autres touches génériques (*Alt*) ou spécifiques à la plate-forme (*Win*, ⌘) permettent aussi de changer de mode. Les touches modifieurs sont également utilisés comme sélecteur de mode du pointeur souris. Typiquement, la touche *Shift* peut-être utilisée pour sélectionner plusieurs objets à la souris. De manière surprenante, nous n'avons trouvé aucune étude proposant de changer la position du délimiteur différenciant entrée de texte et sélection de commandes. Une exception est le mode de changement d'outils dans certaines applications *orientées souris* telles que des logiciels de montage vidéo (Adobe Premiere), de retouche (Adobe Photoshop), de modélisation 3D (Autodesk 3D Studio), etc. Dans ces applications où le besoin d'entrer du texte est relativement rare, l'utilisateur peut sélectionner un outil fréquent en appuyant sur une lettre (sans appuyer sur un modifieur). Le mode texte n'étant alors activé que dans certains contextes particuliers, généralement en fonction du composant graphique qui a le focus à un certain instant. TouchCtrl est une technique d'interaction inspirée par ces raccourcis clavier à activation *implicite*, que nous cherchons à généraliser à un plus grand nombre d'applications incluant les éditeurs de texte dont les tâches de formatage sont souvent réalisées à la souris ou à l'aide de raccourcis clavier.

Sélection de commandes au clavier

Les raccourcis clavier sont des alternatives efficaces à la sélection de commandes à l'aide d'un pointeur [22], en particulier pour les actions répétitives (*e.g.*, copier/coller à répétition) [22, 23, 34]. Plusieurs travaux ont facilité l'utilisation des raccourcis clavier [3, 12, 20, 29, 30, 43]. Par exemple, Grossman *et al.* augmentent l'exposition des raccourcis clavier soit en augmentant la saillance des raccourcis clavier, soit en augmentant le coût de la sélection à la souris [12]. Récemment, Malacria *et al.* ont proposé *ExposeHK* [29], qui affiche les raccourcis clavier au dessus des boutons de la barre d'outil leur correspondant lorsque la touche *Ctrl* est pressée. Ainsi les utilisateurs n'ont pas besoin de se souvenir des raccourcis clavier mais juste de les reconnaître après avoir pressé le modifieur.

Ces systèmes augmentent la performance des utilisateurs en favorisant l'*apprentissage* des raccourcis clavier. Cependant, ces travaux utilisent toujours les touches modifieurs du clavier pour basculer de mode. De plus certains accords requis pour réaliser ces raccourcis clavier peuvent nécessiter des hyper-extensions, hyper-flexions, écartement des doigts difficile à réaliser et donc entraîner une charge musculaire statique inconfortable [37]. À l'inverse, nous étudions le coût d'appuyer sur la touche *Ctrl* pour améliorer l'*exécution* des raccourcis clavier en minimisant la demande physique.

Claviers et souris avancés

Alors que les claviers et souris n'ont pratiquement pas changé conceptuellement ces 30 dernières années, quelques études et produits commerciaux ont récemment proposé des approches différentes [3, 6, 32, 35]. Dans le

contexte de la sélection de commandes, il existe des claviers ayant un petit écran sur chaque touche afin d'y afficher des icônes correspondant à des raccourcis clavier [6, 32, 35]. Des claviers comme celui du Xerox Star [5] ou le Microsoft's Office Keyboard [31] proposent des touches dédiées à des commandes particulières. Cependant ces solutions ne peuvent être appliquées que dans des cas spécifiques, sont limités à des ensembles de commandes réduits et nécessitent de modifier le clavier.

Des études portent aussi sur l'augmentation ou l'utilisation différentes de souris [4, 19, 44, 9]. Par exemple, PadMouse permet à l'utilisateur de sélectionner des commandes en réalisant des gestes sur un touchpad attaché à une souris. Plus spécifique, le système d'exploitation X11 [46, 10] associe par défaut la commande *coller* au bouton du milieu d'une souris, mais cette solution se limite à une unique commande. Cependant ces approches consistent à réaliser la sélection de commande uniquement à la souris, et perdent donc l'expressivité du clavier.

Enfin, certains logiciels permettent de changer les associations entre les touches du clavier ou les boutons de la souris avec les actions à exécuter. Par exemple, des claviers pour joueurs incluant des touches programmables (e.g. Logitech G105 [27]). Le logiciel X-Mouse Button Control permet aussi d'associer les boutons de la souris à des combinaisons arbitraires de touches claviers. Bien que ces logiciels permettent d'implémenter CtrlMouse et TouchCtrl, ils n'informent pas sur l'efficacité des différentes associations. Ce projet vise à explorer une partie de cet immense espace de conception en assignant les modificateurs Ctrl et Shift sur la souris.

Transition de la main entre le clavier et la souris

Ce travail est aussi lié aux techniques qui combinent les interactions avec le clavier et la souris simultanément [41, 6, 38, 15]. Le trackpoint d'IBM [41], un périphérique de pointage isométrique, a été placé au centre du clavier pour éviter les pertes de temps et les distractions. Des travaux plus récents ont remplacé les périphériques de pointage en intégrant des capteurs capacitifs dans le clavier [6, 15]. Enfin Thumbsense conserve le touchpad, mais utilise les touches du clavier comme boutons de souris [38]. Tout comme ces études, notre projet vise à réduire les transitions de la main entre le clavier et la souris. Cependant, contrairement à ces études, nous n'essayons pas de *déplacer* le pointage dans le clavier, mais nous déplaçons les modificateurs sur le périphérique de pointage. En conséquence nous réduisons le besoin de déplacer la main vers le clavier quand les utilisateurs ont la main sur la souris et souhaitent saisir un raccourci clavier.

En résumé, à notre connaissance il n'y a pas de travail sur des méthodes alternatives pour délimiter le mode d'entrée de texte du mode sélection de commandes. La touche Ctrl est placée historiquement à côté de la touche espace sans que ce choix n'ait été remis en cause suite aux évolutions de l'usage des ordinateurs.

ESPACE DE CONCEPTION

Les claviers sont utilisés à la fois pour saisir du texte et invoquer des commandes à l'aide de raccourcis clavier. Les utilisateurs exécutent généralement des commandes en pressant des touches modificateur dédiées telles que la touche Ctrl. Cette approche utilise comme délimiteur une action *explicite* sur le clavier. Nous explorons ici des approches complémentaires et étudions comment la souris peut être une *position* alternative de délimiteur, et présentons deux techniques d'interaction qui diffèrent par leur type d'*activation*. CtrlMouse duplique les délimiteurs classiques du clavier sur les boutons d'une souris localisés sous le pouce, que l'utilisateur peut alors utiliser pour changer *explicitement* de mode. A contrario, TouchCtrl active *implicitement* le mode commande quand la main de l'utilisateur est en contact avec le dispositif de pointage. Ces approches sont résumées en table 1.

		Position	
		Clavier	Souris
Activation	Explicite	Application <i>textuelle</i>	CtrlMouse
	Implicite	Application <i>graphique</i>	TouchCtrl

Table 1: Différentes alternatives pour passer du mode entrée de texte au mode commande. Les applications *textuelles* utilisent une action *explicite* de l'utilisateur pour changer le mode du clavier. Les applications *graphiques* disposent de raccourcis *implicites*, sans modificateurs. CtrlMouse duplique les modificateurs sur la souris et demande une action *explicite* de l'utilisateur. TouchCtrl change *implicitement* de mode quand l'utilisateur a sa main sur la souris.

CtrlMouse

CtrlMouse est une technique d'interaction qui facilite l'utilisation des raccourcis clavier quand la main de l'utilisateur est située sur le dispositif de pointage. CtrlMouse duplique les délimiteurs classiques du clavier sur les boutons (localisés sous le pouce) d'une souris (Figure 1). Le mode du clavier passe d'*entrée de texte* à *commande* dès que l'utilisateur presse un de ces boutons, et retourne au mode *entrée de texte* quand ces boutons sont relâchés. Comme les touches modificateur sont dupliquées sur la souris, et peuvent être activées avec la main dominante, l'utilisateur n'a plus besoin d'effectuer d'accords sur le clavier, ni de déplacer sa main dominante sur le clavier pour saisir des raccourcis compliqués. Plus précisément, dupliquer les touches de modification sur la souris présente les avantages suivants :

Expressivité. CtrlMouse ne remplace pas les touches de modification du clavier, mais offre une solution alternative à l'utilisateur. Ainsi, l'utilisateur peut choisir d'utiliser les touches de modification qu'il souhaite (souris ou clavier) en fonction de ses préférences.

Coordination des doigts. Les raccourcis clavier traditionnels nécessitent de réaliser des accords avec les doigts. Selon la position des touches et du nombre de modificateurs, les raccourcis clavier peuvent nécessiter une coordination des doigts subtile pour leur exécution, ainsi qu'une charge musculaire statique inconfortable [37]. En divisant les modificateurs et les touches de symbole, CtrlMouse réduit la coordination des doigts en utilisant les deux mains pour réaliser l'exécution des raccourcis.

Association des raccourcis clavier. D'une manière générale, les raccourcis clavier sont associés à une touche du clavier qui va reposer sur une association mnémotecnique pour faciliter la mémorisation. Cependant, les raccourcis clavier fréquents tendent également à être positionnés sur la partie gauche du clavier pour réduire la distance avec la touche `Ctrl` gauche. En conséquence, il est parfois nécessaire d'augmenter le nombre de modificateurs pour les maintenir dans cette zone. Comme `CtrlMouse` réduit les contraintes d'accords, les raccourcis claviers peuvent être distribués de manière plus uniforme sur le clavier. Cela donne plus de flexibilité aux concepteurs pour définir une association de commande efficace et réduire les raccourcis nécessitant plusieurs modificateurs [3].

Plusieurs modificateurs. Quand il y a plusieurs modificateurs, `CtrlMouse` offre un avantage significatif. Il permet d'activer deux modificateurs à la fois en utilisant un appui unique du pouce sur les deux boutons simultanément.

Transitions souris/clavier. Certains raccourcis clavier sont difficiles à exécuter à cause de la position de la touche symbole, en particulier si elle est située sur la partie droite du clavier [3]. Alors que les utilisateurs peuvent réaliser des raccourcis clavier avec la touche `Ctrl` de la partie droite du clavier, nous avons observé que dans les applications orientées souris, ils conservent leur main gauche sur la touche `Ctrl` gauche et utilisent leur main droite pour la touche symbole. Les boutons de `CtrlMouse` suppriment ces transitions Souris/Clavier avec la main droite.

Compatibilité avec fonctionnalités existantes `CtrlMouse` écoute l'état des boutons de la souris situés sous le pouce uniquement lorsqu'une touche du clavier est appuyée. Cela signifie que `CtrlMouse` est compatible avec les actions usuelles des boutons de la souris situés sous le pouce comme *précédent* et *suivant* pour naviguer dans l'historique d'un navigateur.

Implémentation matérielle. `CtrlMouse` ne nécessite pas de modifications matérielles car plusieurs souris commerciales (e.g., Logitech M705 [25] ou Logitech M905 [26]) ont déjà deux boutons sous le pouce. Techniquement un appui inférieur à 200ms déclenche les actions usuelles, et un appui plus long lorsque une touche du clavier est enfoncée correspond au maintien d'un modificateur.

En résumé, `CtrlMouse` diffère des raccourcis claviers traditionnels par la *position* du modificateur : les utilisateurs peuvent presser des boutons dédiés sur la souris au lieu de touches modificateur du clavier. Cependant, l'utilisation d'une souris comme nouvelle modalité de basculement en mode sélection de commande offre de nouvelles possibilités illustrées par `TouchCtrl`.

TouchCtrl

TouchCtrl est une variante de `CtrlMouse`, limitée à un seul modificateur (typiquement `Ctrl`) mais qui ne requiert pas d'appui *explicite*. Elle est basée sur l'hypothèse que les utilisateurs n'utilisent pas ou peu le clavier pour saisir du texte quand la main droite tient la souris. `TouchCtrl` bascule *implicitement* le clavier en mode commande dès que la main droite est sur la souris. `TouchCtrl` a les mêmes avantages que `CtrlMouse` (meilleure coordina-

tion des doigts, moins de déplacements, raccourcis plus simples à associer et moins de transitions clavier/souris) mais requiert également moins d'efforts de la main droite.

Compatibilité. `TouchCtrl` se base sur l'hypothèse que les utilisateurs ne saisissent pas ou peu de texte quand la main droite est sur la souris. Comme nous l'avons observé, cela correspond en effet à des cas rares. Cependant, s'ils souhaitent saisir du texte (quelques mots dans un champ de recherche par exemple), il leur suffit de lever légèrement la main de la souris pour saisir le texte de la main gauche.

Plusieurs modificateurs. `TouchCtrl` est plus sensible aux modificateurs multiples que `CtrlMouse` car l'information binaire (main en contact avec la souris ou non) ne permet de remplacer qu'un seul modificateur. Il est donc toujours nécessaire de presser explicitement un modificateur supplémentaire (sur le clavier, ou éventuellement sur la souris dans un contexte qui combinerait `TouchCtrl` et `CtrlMouse`) pour effectuer un raccourci clavier à plusieurs modificateurs. Cependant cette technique d'interaction permet à l'image de `CtrlMouse` d'encourager les concepteurs à utiliser plus de raccourcis à modificateur unique en exploitant plus la partie droite du clavier.

Implémentation matérielle. `TouchCtrl` est supporté par les souris multitouch du commerce comme Apple MagicMouse [1] ou Microsoft TouchMouse [33]. De plus, `TouchCtrl` peut aussi être utilisée avec un touchpad et activer les commandes quand au moins un doigt est en contact avec la surface.

Relocaliser les modificateurs sur la souris peut apporter des bénéfices à la fois qualitatifs et quantitatifs par rapport au clavier. Dans les sections suivantes, nous présentons deux expériences contrôlées et un déploiement préliminaire en situation écologique pour mettre en avant les bénéfices de `CtrlMouse` et `TouchCtrl`.

ÉTUDE 1 : ADOPTION

Cette étude vise à étudier et comprendre l'impact de la *demande d'entrée de texte* (DET) d'une application sur les performances de `CtrlMouse` et `TouchCtrl`. Intuitivement, les applications *graphiques* nécessitant beaucoup d'actions au curseur (DET nulle) exploitent au mieux les avantages de nos techniques car la main est déjà posée sur la souris. Au contraire, `CtrlMouse` et `TouchCtrl` semblent moins adaptées aux éditeurs de texte (forte DET) car les participants doivent entrer du texte avant chaque sélection de commandes, les deux mains se retrouvant alors positionnées sur le clavier. Cependant, nous ne savons pas à partir de quelle demande d'entrée de texte, les utilisateurs vont préférer adopter une technique plutôt qu'une autre et s'ils sont prêts à abandonner les raccourcis claviers classiques.

Methodologie

Participants et équipement. Nous avons recruté via listes de diffusions douze participants (3 femmes), âgés entre 24 et 35 ans (*moyenne* = 27,8, σ = 3,9). Tous étaient familiers avec les raccourcis clavier. L'expérience a été réalisée sur un PC sous Windows 7 avec un écran LCD (960 × 1280 pixels). Le clavier était un clavier standard DELL AZERTY. Nous avons utilisé une souris standard

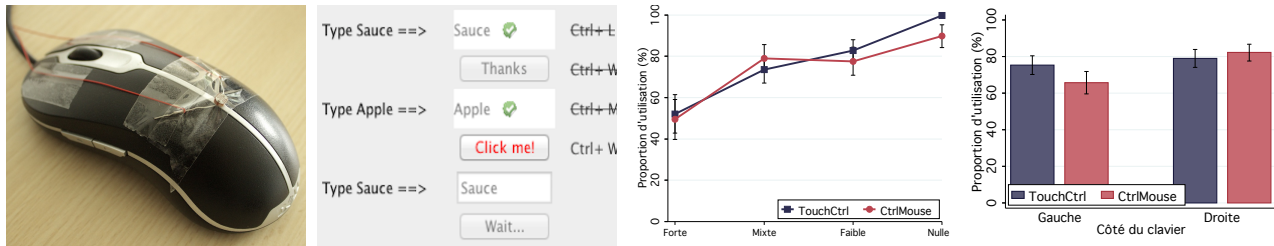


Figure 2: Souris avec des boutons sous le pouce et une photorésistance sur le dos (gauche); Fenêtre expérimentale pour la condition mixte (milieu-gauche). Proportion d'utilisation des modificateurs sur la souris par DET (milieu-droite) et par côté du clavier (droite)

DELL avec deux boutons sous le pouce, et augmentée d'une photorésistance pour capter la main de l'utilisateur lorsqu'elle se trouvait dessus (Figure 2, gauche). La photorésistance était interfacée avec Arduino.

Tâche. La tâche est adaptée de la *Home task* de [31] et consiste à faire varier la *demande d'entrée de texte*. Pour chaque essai, les utilisateurs doivent éventuellement saisir du texte, cliquer sur un bouton avec la souris puis exécuter un raccourci clavier affiché dans un label. Le bouton, le(s) champ(s) de texte et le label s'affichaient en début d'essai (Figure 2, milieu-gauche). Ce dispositif expérimental simule le comportement d'un utilisateur expert qui saisit du texte, effectue une sélection et invoque une commande qu'il connaît à l'avance. L'essai se termine et passe au suivant une fois que le raccourci clavier saisi et tous les modificateurs sont relâchés. Ceci empêche un utilisateur de maintenir un modificateur pendant toute l'expérience. Un son informe les utilisateurs lorsqu'ils ont fait une erreur de sélection.

Nous avons simulé l'entrée de texte en demandant à l'utilisateur de saisir des mots simples (sauce, cloak ou apple) avec les deux mains. Ces mots utilisent des lettres des deux côtés du clavier, ce qui assure que les utilisateurs utilisent leurs deux mains lors de l'entrée de texte. Dans la condition à forte demande d'entrée de texte, le pointage servait uniquement à donner le focus au prochain champ de texte dans lequel taper du texte. Nous avons ignoré la condition où les participants ne réalisent aucune tâche de pointage (et entraînent uniquement du texte) car les utilisateurs n'auraient aucune raison d'abandonner les raccourcis clavier.

Demande d'entrée de texte (DET). Il y avait quatre niveaux différents :

- forte : tous les essais nécessitent d'entrer du texte avant de sélectionner une commande. L'utilisateur devait néanmoins toujours utiliser la souris pour donner le focus à un champ de texte avant de pouvoir y entrer du texte dedans.
- mixte : la moitié des essais nécessitent d'entrer du texte avant de saisir la commande.
- faible : un tiers des essais nécessitent d'entrer du texte avant de sélectionner la commande.
- nulle : aucun essai ne nécessite de saisir du texte avant de sélectionner une commande.

Technique. Nous avons comparé les deux techniques d'interaction CtrlMouse et TouchCtrl.

Raccourcis claviers. Les raccourcis clavier utilisaient un seul modificateur (Ctrl). La moitié $\{w, x, f\}$ des 6 lettres utilisées étaient situées sur le côté gauche du clavier. L'autre moitié $\{p, l, m\}$ était située sur le côté droit. Ceci permet d'étudier l'impact du facteur *côté du clavier* sur l'adoption des techniques.

Procédure. Les expérimentateurs expliquaient d'abord la tâche et les deux nouvelles techniques d'interaction (CtrlMouse et TouchCtrl) aux participants. Ces derniers avaient pour instruction de réaliser la tâche aussi rapidement et précisément que possible. Il était indiqué également que les participants étaient libres de réaliser les sélections avec les raccourcis clavier classiques ou avec la nouvelle technique proposée dans la condition. Ainsi avec CtrlMouse ils pouvaient soit presser le modificateur sur la souris soit sur le clavier. Avec TouchCtrl ils pouvaient soit mettre leur main sur la souris et réaliser le raccourci sans retirer leur main de la souris ou utiliser la touche Ctrl du clavier. Les essais étaient chronométrés depuis l'apparition du stimulus jusqu'à ce que la commande soit saisie.

Plan expérimental. L'expérience suivait un plan à mesures répétées $2 \times 8 \times 4 \times 2$ avec les facteurs intra-sujets *Technique* (CtrlMouse et TouchCtrl), *Bloc* (niveaux 1-8), *demande d'entrée de texte (DET)* (forte, mixte, faible, et nulle) et *côté du clavier* (droite ou gauche). L'ordre des *Techniques* et *DET* était balancé entre les participants à l'aide d'un carré latin. Chaque bloc était composé de 3 essais pour chaque *côté du clavier*. Les raccourcis claviers étaient présentés dans un ordre aléatoire. La principale mesure est la proportion de raccourcis claviers classiques utilisés pour chaque condition. Un essai était considéré comme effectué au clavier si la touche Ctrl du clavier était pressée au moment de sélection de la commande. L'expérience durait à peu près 20 minutes avec un total de 384 essais par participant. Les participants étaient libres de prendre une pause tous les deux blocs.

Resultats

La proportion d'utilisation de CtrlMouse et TouchCtrl globale était de 75.5% ($\sigma = 37$). La Figure 2 résume les proportions de sélections de commande utilisant les modificateurs sur la souris pour chaque *Technique* en fonction de la *demande d'entrée de texte* (milieu droite) et du côté du clavier (droite). Une analyse de variance montre un effet significatif sur les facteurs suivants : *DET* ($F_{3,33} = 11.43, p < .001$), *Block* ($F_{3,33} = 3.1, p < .05$) et *Côté du clavier* ($F_{1,11} = 8.67, p < .05$).

Les résultats confirment que la proportion d'utilisation des modificateurs sur la souris est inversement proportionnelle

à la demande d'entrée de texte. Celle-ci est de 50.8% ($\sigma = 46.8$) pour une demande *forte* et est significativement inférieure à une demande *mixte* (76.2%, $\sigma = 32.4$), une demande *faible* (80.2%, $\sigma = 29.6$) et une demande *nulle* (94.8%, $\sigma = 19.6$). Nous n'avons pas détecté de différence significative entre les trois dernières demandes. Ces résultats montrent que les participants adoptent nos deux techniques alors qu'ils étaient bien plus entraînés avec les raccourcis claviers classiques. Ils adoptent Ctrl-Mouse et TouchCtrl d'autant plus que la demande d'entrée de texte est faible.

La position des raccourcis claviers (*côté du clavier*) influence également l'adoption de nos techniques. Ctrl-Mouse et TouchCtrl sont significativement plus utilisés lorsque les raccourcis sont sur la droite du clavier (80.5%, $\sigma = 33.4$) plutôt que sur la gauche du clavier (70.5%, $\sigma = 39.8$). Ce résultat s'explique probablement par le fait que les raccourcis claviers classiques avec une lettre sur la droite du clavier nécessitent des aller-retours fastidieux entre la souris et le clavier.

ETUDE 2 : PERFORMANCE

Cette étude vise à mieux comprendre les différences entre CtrlMouse et TouchCtrl dans le contexte où ces techniques s'avèrent le plus utiles, c'est à dire un contexte avec une faible demande d'entrée de texte (voir étude précédente). Nous étudions en particulier l'influence de la *Position* du délimiteur (*Clavier* vs. *Souris*), du mode d'activation (CtrlMouse vs. TouchCtrl) et du *nombre de modifieurs* (1 ou 2) sur les performances.

Cette expérience est similaire à la précédente avec quelques modifications décrites ci-dessous :

- Tâche. Les utilisateurs n'avaient pas de texte à saisir (voir Figure 3, gauche). L'objectif est de simuler des tâches réalisées à la souris interrompues par des raccourcis clavier.
- Raccourcis claviers. Les participants utilisaient soit un (Ctrl) soit deux modifieurs (Ctrl+Shift)
- Techniques. Les participants ne testaient qu'une seule modalité à la fois car nous nous intéressions ici à la performance de chaque technique et non à l'adoption de celles-ci par les utilisateurs. Les participants ne pouvaient donc pas utiliser les touches Ctrl et Shift du clavier pendant les conditions CtrlMouse et TouchCtrl. Inversement les boutons de la souris étaient désactivés dans la condition *Clavier*.

12 participants (3 femmes) ont été recrutés pour cette expérience.

L'expérience suivait un plan à mesures répétées $3 \times 8 \times 2 \times 2$ avec les facteurs intra-sujets : *Technique* (Ctrl-Mouse, TouchCtrl et Clavier), *Bloc* (niveaux 1-8), *Nombre de modifieurs* (Ctrl and Ctrl+Shift) et *Côté du clavier* (Gauche et Droit). L'ordre de *Technique* était balancé parmi les participants à l'aide d'un carré latin. Il y avait 10 blocs par techniques (dont 2 pour l'entraînement écartés de l'analyse). Chaque bloc comprenait 12 essais (6 lettres x 1 ou 2 modifieurs). Les variables dépendantes sont le temps de complétion d'un essai (essais avec erreurs supprimés), le taux d'erreur, la charge de travail (mesurée

avec un questionnaire NASA-TLX [16]) et les préférences utilisateurs. L'expérience durait environ 20 minutes, avec un total de 288 essais par participant.

Resultats

Le temps moyen pour réaliser un essai était de 1.82s ($\sigma = 0.59$), avec un taux d'erreur de 3%. La Figure 3 résume les temps de complétion pour les 3 *Techniques* par *Bloc* (milieu-gauche), *Côté du clavier* (milieu-droite) et *Nombre de modifieurs* (droite). Une analyse de variance montre un effet significatif pour tous les facteurs : *Technique* ($F_{2,22} = 9,46, p <,001$), *Bloc* ($F_{7,77} = 9,05, p <,001$), *Côté du clavier* ($F_{1,11} = 11,95, p <,01$) et *Nombre de modifieurs* ($F_{1,11} = 47,9, p <,001$).

Le temps moyen de complétion d'un essai avec TouchCtrl (1,68s, $\sigma = 0,51$) était significativement plus rapide que CtrlMouse (1,88s, $\sigma = 0,52$; gain de 12,5%) et les raccourcis clavier (1,89s, $\sigma = 0,69$; gain de 11,9%). Cependant, il n'y a pas de différence significative entre Ctrl-Mouse et les raccourcis clavier. La sélection de raccourcis avec un seul modifieur (1,68s, $\sigma = 0,46$) était significativement plus rapide que les raccourcis à deux modifieurs (1,96s, $\sigma = 0,66$; gain de 16,5%).

Il y avait aussi une interaction *Technique* \times *Modifieur* ($F_{2,22} = 6,09, p <,01$, Figure 3 (droite), causée par un temps de sélection plus lent pour les commandes à deux modifieurs avec TouchCtrl et les raccourcis clavier. L'analyse post-hoc montre une différence significative entre la sélection à un seul modifieur avec TouchCtrl (1,47s, $\sigma = 0,37$) et toutes les autres conditions. Il est intéressant de noter que CtrlMouse est la seule technique pour laquelle il n'est pas significativement plus long de sélectionner des commandes à deux modifieurs qu'à un seul.

Enfin le temps de sélection moyen global pour les touches du côté gauche du clavier (1,76s, $\sigma = 0,55$) était significativement plus court que pour les touches du côté droit (1,87s, $\sigma = 0,62$), mais le gain (6%) demeure faible.

Préférences subjectives

Nous avons analysé les questionnaires NASA TLX avec des tests de Kruskal Wallis. Seule la *demande physique* a montré un effet significatif entre les *Techniques* ($\chi^2 = 14,8, p <,001$). Les comparaisons paires de Wilcoxon montrent que TouchCtrl (4,5/100, $\sigma = 10,2$) a une demande physique significativement plus faible que CtrlMouse (13/100, $\sigma = 16,9$) et les raccourcis clavier (39/100, $\sigma = 19$). La différence entre CtrlMouse et les raccourcis clavier était aussi significative.

Cette observation contribue à valider nos choix de conception. Les participants nous ont confirmé lors des entretiens après l'expérience que les sources principales de fatigue sont les allers retours entre le clavier et la souris et la charge musculaire statique. La charge physique moindre de CtrlMouse et TouchCtrl est un signe d'amélioration.

Discussion

Les résultats de cette expérience montrent que l'activation implicite du mode de sélection de commande à la souris avec TouchCtrl a un intérêt. Bien que la différence temporelle par sélection peut sembler petite (0,21s), le

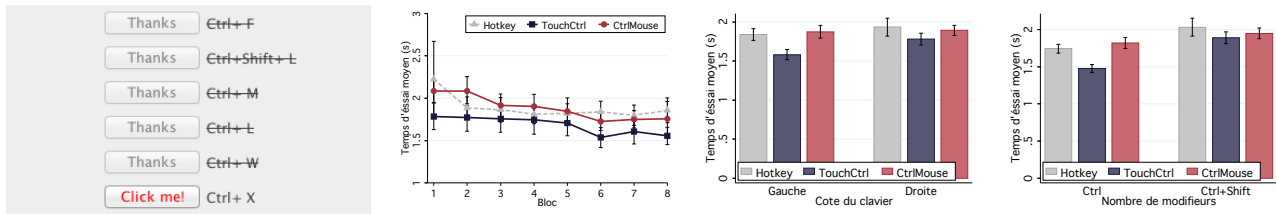


Figure 3: Fenêtre de l'expérience (gauche) : Les participants devaient cliquer sur un bouton, puis réaliser une sélection de commande. Résultats : temps moyen de complétion par bloc (milieu-gauche), temps moyen de complétion par côté du clavier (milieu-droit) et temps moyen de complétion par nombre de modifieurs (droite).

gain est important (11,9%) et s'avère utile lorsque les utilisateurs sélectionnent un grand nombre de commandes. Cette différence est proche de la prédiction par le modèle GOMS [8] si on considère la différence comme un appui sur une touche séparée (0,28s). TouchCtrl est donc plus rapide que les raccourcis clavier les plus courants (avec un seul modifieur). De plus CtrlMouse et TouchCtrl ne pénalisent pas la sélection de commandes avec deux modifieurs.

Le manque de différence significative entre CtrlMouse et les raccourcis claviers peut paraître surprenante. Cependant, CtrlMouse demande toujours une action *explicite* de l'utilisateur pour activer les modifieurs, action à laquelle nos participants n'étaient absolument pas entraînés au début de l'expérience (contrairement aux raccourcis clavier). Par conséquent, il est possible que plus d'entraînement ait été nécessaire avec CtrlMouse pour atteindre le seuil de performance de nos participants.

D'autre part, nos résultats n'ont pas révélé d'interaction *Technique* \times *Côté du clavier*, ce qui est assez surprenant. D'un point de vue théorique CtrlMouse (et dans d'autres proportions, TouchCtrl dans le cas d'une commande à un seul modifieur) devraient être aussi rapides pour les touches sur la gauche et la droite du clavier. En effet, il n'a plus à reposer sa main sur le côté gauche du clavier pour accéder à la touche `Ctrl` et pourrait reposer sa main au centre du clavier et ainsi optimiser la distance avec les différentes touches. Nous avons cependant observé que nos participants continuaient à positionner leur main gauche sur la partie gauche du clavier, probablement par habitude.

Notre expérience n'a pas soulevé de différence significative sur la performance de CtrlMouse, quel que soit le nombre de modifieurs (1 ou 2), contrairement à TouchCtrl et raccourcis claviers traditionnels. Cette observation peut s'expliquer par le fait que les deux modifieurs sont positionnés sur la souris, sous le pouce, et que les actions moteurs sont quasi-identiques qu'il s'agisse d'appuyer sur un seul ou les deux boutons.

Nous pouvons également discuter ces résultats au regard des études portant sur l'interaction bi-manuelle. De nombreux travaux s'appuient sur la théorie de la chaîne cinématique de Guiard [13] : « la main non-dominante précède la main dominante et fournit le contexte de l'action ». Nos techniques ne suivent pas ce principe car la main droite définit d'abord le contexte sur la souris avant que la main gauche agisse sur le clavier en appuyant sur une touche. En ne s'appuyant pas sur le principe de la chaîne cinématique, on pouvait craindre un effort cognitif plus

important et donc un temps d'exécution plus long. Cependant, les résultats positifs tendent à montrer que la baisse de la complexité bio-mécanique (en simplifiant les mouvements de la main gauche) est plus importante que la hausse de la complexité cognitive. D'autres expériences devront être entreprises pour quantifier précisément l'effort cognitif associé à nos techniques.

Ces expériences en laboratoire nous informent sur l'impact de la localisation et du type d'activation des modifieurs, l'impact de la demande d'entrée de texte ainsi que sur le coût du nombre de modifieurs. Cependant, elles ne nous informent pas sur le comportement des utilisateurs en situation réelle. La section suivante apporte quelques éléments de réponse préliminaires sur cette question.

DEPLOIEMENT

Nous avons implémenté CtrlMouse et TouchCtrl afin de (1) faciliter leur intégration dans les systèmes existants et (2) confirmer de manière écologique les résultats positifs obtenus lors des évaluations en laboratoire. Dans cette section nous décrivons l'implémentation des deux techniques d'interaction et les premiers résultats qualitatifs de leur déploiement sur le terrain.

Implémentations

TouchCtrl

TouchCtrl est une application pour MacOSX qui s'exécute en tâche de fond et doit être utilisé avec un dispositif de pointage multi-points d'Apple comme la Magic Mouse[1] ou le Magic Trackpad[28]. Elle a été réalisée en Objective-C sous les API Cocoa et Carbon et repose sur deux composants logiciels principaux : (1) détection du contact de la main sur la souris et (2) émulation de l'exécution des raccourcis claviers.

Notre implémentation utilise le framework privé d'Apple *MultitouchSupport* pour identifier si la main est en contact avec la souris. Ce framework permet à un programme d'accéder à tout instant à la liste des points en contact (position, taille et orientation du point de contact) avec un dispositif de pointage multi-points d'Apple. Notre implémentation considère que la main est en contact avec la souris si au moins un point de contact existe.

Nous avons utilisé les *event taps* d'Apple pour émuler l'exécution des raccourcis claviers. Notre implémentation intercepte les *event taps* d'appui de touches et modifie sa propriété champs *modifierflags* pour prétendre que la touche `⌘` est pressée si la main est en contact avec la souris. Nous injectons alors le champs modifié dans les événements qui sont renvoyés au système. Ceci a pour effet d'activer le raccourci clavier correspondant et d'éviter

de simuler que la touche \mathfrak{H} est pressée en permanence, typiquement pour éviter de modifier les clics souris.

CtrlMouse

CtrlMouse peut être déployé avec n'importe quelle souris ayant au moins deux boutons à proximité du pouce. Nous avons implémenté la technique sous Windows avec Auto-Hotkey [2]. Cet outil permet de manipuler les événements d'entrée, et en particulier de les rediriger et en générer. Ainsi nous déclenchons les événements de la touche `Ctrl` en réaction aux événements du premier bouton du pouce de la souris et `Shift` sur le deuxième.

Cet outil permet soit de remplacer le comportement habituel des touches et boutons, soit d'y ajouter des actions. Certains utilisateurs souhaitent conserver leur utilisation classique, par exemple pour passer aux pages précédentes et suivantes dans un navigateur. D'autres ne souhaitent pas ce comportement. Nous avons donc réalisé deux implémentations.

Resultats qualitatifs du déploiement

Nous avons déployé nos techniques auprès de 6 utilisateurs (4 CtrlMouse et 2 TouchCtrl) pendant 2 mois. Les applications étaient lancées au démarrage de la session.

Les utilisateurs ont globalement apprécié le fait de pouvoir effectuer `Ctrl` et `Shift` sur la souris, surtout au fur et à mesure qu'ils découvraient ce qu'ils pouvaient en faire. Un utilisateur de CtrlMouse et un de TouchCtrl soulignent le confort pour les raccourcis à deux modificateurs. Deux participants disent apprécier l'utilisation des modificateurs sur la souris de CtrlMouse pour réaliser des sélections multiples. Un utilisateur de TouchCtrl apprécie pouvoir copier et coller des fichiers dans Finder à l'aide d'une simple touche, et plusieurs suggèrent l'intérêt de CtrlMouse pour le contrôle de paramètres de commandes à la souris (aimantation sur grille, contraintes, zoom, ajouter/supprimer à la sélection). Ce potentiel n'était pas exploité par tous les utilisateurs, deux de nos utilisateurs n'utilisant pratiquement pas d'application orientées graphique.

Faux positifs. L'intérêt de TouchCtrl est l'activation implicite du mode commande quand la main est sur la souris. Cependant les actions implicites possèdent des risques de faux positifs. Nos utilisateurs ont signalé avoir rencontré le problème quelques fois. Il est important d'éviter les faux positifs sur les commandes critiques (\mathfrak{H} -Q) et de pouvoir facilement recouvrir des autres. Ainsi les commandes critiques doivent conserver une activation explicite, par exemple avec un modificateur sur le clavier ou la souris. La posture kinesthésique servant à maintenir le quasi-mode n'évoque pas ce changement de mode, car les utilisateurs sont habitués à cette posture et à ce qu'elle n'influe pas sur leurs interactions. Il faut donc veiller à fournir un retour visuel ou haptique supplémentaire, suggérant le mode courant. Cependant, un de nos utilisateurs a déclaré s'être habitué à terme à TouchCtrl.

Implémentation. Les utilisateurs ont soulevé des problèmes d'implémentation qui ont influencé l'utilisation de ces techniques. Deux utilisateurs de CtrlMouse ont désactivé fréquemment le système car ils utilisent les

boutons du pouce de la souris pour naviguer dans l'historique de leur navigateur web. L'implémentation permet de réaliser les deux fonctionnalités en même temps, cependant la page change quand les boutons sont utilisés. Une implémentation idéale ne déclencherait les modificateurs si et seulement si aucune action n'est associée aux boutons au moment où ils sont utilisés.

Matériel. Un utilisateur a signalé que les boutons du pouce de sa souris (Logitech M500) ne sont pas à la position de repos du pouce. Cela permet d'éviter des activations accidentelles avec une utilisation classique de ces boutons. Cependant dans notre cas ces boutons activent des quasi-modes et non des commandes, donc ce risque est moindre. Avoir les boutons dans une position confortable reste souhaitable pour des sélections multiples, qui nécessitent des clics gauche avec l'index et des mouvements de la souris en plus du maintien d'un bouton du pouce.

Ce déploiement nous a appris tant pour l'utilisation que pour l'implémentation. Au niveau de l'utilisation, les utilisateurs ont commencé à s'approprier les techniques et à y trouver leurs propres usages, au delà des raccourcis clavier, ce qui est particulièrement encourageant. Nous avons aussi constaté que l'implémentation n'était pas aussi triviale que prévue, ce qui est difficile à déceler avec des expériences en laboratoire. Nous avons mis en avant des problèmes ergonomiques qui peuvent nuire à l'utilisabilité des techniques, ainsi que des solutions pour y remédier.

DISCUSSION

Nous discutons dans cette section d'autres usages potentiels de CtrlMouse et TouchCtrl.

Application aux touchpads : TouchCtrlPad

De plus en plus d'utilisateurs d'ordinateurs, et en particulier de portables, utilisent un touchpad. Les études présentées plus haut ont montré certains avantages à l'utilisation de CtrlMouse ou TouchCtrl avec une souris. Cependant, il n'était pas évident que l'on puisse transposer ces techniques aux touchpads.

Inspirés par [38, 18], nous proposons TouchCtrlPad. Lorsque les utilisateurs posent un pouce sur le pavé numérique, le mode `Ctrl` est activé. Les autres doigts sont alors libres pour appuyer la lettre d'un raccourci clavier. Cette technique a plusieurs avantages. Tout d'abord, activer le mode `Ctrl` ne nécessite pas de flexion du pouce, ce qui est potentiellement moins contraignant que l'utilisation de les touches `Ctrl` ou \mathfrak{H} . Ensuite, tous les doigts longs peuvent rester sur la seconde rangée de lettres, la position par défaut. C'est une différence avec les raccourcis claviers sous Windows, pour lesquels la touche `Ctrl` est souvent utilisée avec l'auriculaire alors que la touche \mathfrak{H} sous Mac est plus utilisée avec le pouce. Cependant, même sous Mac, selon la position de la touche symbole du raccourci, la main peut être amenée à tourner dans une position inconfortable. Enfin, sur un touchpad multi-points, l'utilisateur peut poser un ou deux doigts pour différencier le mode `Ctrl` et le mode `Ctrl+Shift`.

Nous avons implémenté TouchCtrlPad sur MacOS. Les premiers tests effectués par les auteurs pour exécuter des

raccourcis claviers étaient positifs. Cependant, de nombreuses activations accidentelles ont été observées lors des premiers déploiements. La raison est que la position de repos des pouces est très proche et parfois sur les coins du pavé numérique. Pour véritablement bénéficier des avantages de cette technique, il s'agira de filtrer les événements du touchpad venant des coins supérieurs pour éviter les activations accidentelles. Une autre solution consisterait à revoir la position des touches modificateur sur le clavier.

Interactions avancées

L'objectif de notre étude était de comprendre l'impact de la position et du type d'activation des délimiteurs de mode du clavier. Nous avons donc conçu et expérimenté CtrlMouse et TouchCtrl séparément. Cependant il est tout à fait envisageable de combiner CtrlMouse et TouchCtrl. Par exemple `Ctrl` serait activé en posant la main sur la souris, et les deux boutons dupliqueraient les comportements de `Shift` et `Alt`.

Au delà de la sélection de commandes

Les touches `Ctrl` et `Shift` ne sont pas seulement utilisées pour la sélection de commande. Elles permettent aussi de contrôler des paramètres de commandes réalisées à la souris. Suite au déploiement de notre technique, nous avons observé des usages de CtrlMouse et TouchCtrl pour ce genre d'application. Par exemple lors du déplacement d'un objet sur le bureau ou dans un dossier, les modificateurs servent à basculer entre les modes de copie, déplacement ou création de lien. Avec un outil de loupe, on peut choisir entre zoom et dé-zoom. Il est aussi possible d'inclure ou d'exclure des éléments d'une sélection multiple. Toutes ces interactions utilisent typiquement deux mains : une pour la manipulation du pointeur et une pour la sélection des modificateurs. CtrlMouse et TouchCtrl permettent de réaliser ces opérations avec une seule main.

CONCLUSION

Nous avons étudié l'impact de la *position* et du *mode d'activation* des délimiteurs de modes d'entrée de texte et de sélection de commandes. Pour cette étude nous avons présenté CtrlMouse et TouchCtrl, deux techniques d'interaction qui dupliquent les touches modificateurs du clavier sur la souris. CtrlMouse utilise deux boutons sous le pouce qui dupliquent les touches `Ctrl` et `Shift` lorsque l'utilisateur les presse *explicitement*. TouchCtrl active de manière *implicite* le modificateur `Ctrl` lorsque la main est posée sur la souris. Nous avons présenté deux expériences expérimentales. La première montre que l'adoption de la technique est supérieure quand la tâche nécessite peu d'entrée de texte. La deuxième montre que TouchCtrl procure un gain de performance par rapport aux raccourcis claviers classique, et que CtrlMouse est efficace lorsqu'il y a plusieurs modificateurs. Nous avons déployé ces techniques sur une longue durée chez 6 utilisateurs. Nous avons observé des comportements intéressants au delà de la sélection de commande, telle que la manipulation de paramètres de commandes effectuées à la souris avec une seule main. Nous avons aussi détecté des problèmes ergonomiques pouvant impacter l'utilisabilité de ces techniques, et proposé des solutions.

BIBLIOGRAPHIE

1. Apple Magic Mouse. <http://www.apple.com/magicmouse/>.
2. AutoHotkey. <http://www.autohotkey.com/>.
3. Bailly G., Pietrzak T., Deber J. & Wigdor D. Métamorphe: Augmenting hotkey usage with actuated keys. In *CHI '13* (2013), 563–572.
4. Balakrishnan R. & Patel P. The padmouse: facilitating selection and spatial positioning for the non-dominant hand. In *CHI '98* (1998), 9–16.
5. Bewley W. L., Roberts T. L., Schroit D. & Verplank W. L. Human factors testing in the design of xerox's 8010 star office workstation. In *CHI '83* (1983), 72–77.
6. Block F., Gellersen H. & Villar N. Touch-display keyboards: transforming keyboards into interactive surfaces. In *CHI '10* (2010), 1145–1154.
7. Buxton W. Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.* 17, 1 (1983), 31–37.
8. Card S., Newell A. & Moran T. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., 1983.
9. Cechanowicz J., Irani P. & Subramanian S. Augmenting the mouse with pressure sensitive input. In *CHI '07* (2007), 1385–1394.
10. Chapis O. & Roussel N. Copy-and-paste between overlapping windows. In *CHI '07* (2007), 201–210.
11. Evans A. & Wobbrock J. Taming wild behavior: the input observer for text entry and mouse pointing measures from everyday computer use. In *CHI '12* (2012), 1947–1956.
12. Grossman T., Dragicevic P. & Balakrishnan R. Strategies for accelerating on-line learning of hotkeys. In *CHI '07* (2007), 1591–1600.
13. Guiard Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19 (1987), 486–517.
14. Guimbretière F. & Winograd T. Flowmenu: combining command, text, and data entry. In *UIST '00* (2000), 213–216.
15. Habib I., Berggren N., Rehn E., Josefsson G., Kunz A. & Fjeld M. Dgts: Integrated typing and pointing. In *INTERACT '09* (2009), 232–235.
16. Hart S. & Staveland L. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human mental workload* (1988), 139–183.
17. Hinckley K., Baudisch P., Ramos G. & Guimbretiere F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *CHI '05* (2005), 451–460.
18. Hinckley K., Czerwinski M. & Sinclair M. Interaction and modeling techniques for desktop two-handed input. In *UIST '98* (1998), 49–58.
19. Kim S., Kim H., Lee B., Nam T.-J. & Lee W. Inflatable mouse: volume-adjustable mouse with air-pressure-sensitive input and haptic feedback. In *CHI '08* (2008), 211–224.
20. Krisler B. & Alterman R. Training towards mastery: overcoming the active user paradox. In *NordiCHI '08* (2008), 239–248.
21. Kurtenbach G. & Buxton W. Issues in combining marking and direct manipulation techniques. In *UIST '91* (1991), 137–144.
22. Kurtenbach G. P. *The design and evaluation of marking menus*. PhD thesis, Toronto, Canada, 1993.
23. Lane D., Napier A., Peres C. & Sandor A. The Hidden Costs of Graphical User Interfaces: The Failure to Make the Transition from Menus and Icon Tool Bars to Keyboard Shortcuts. *IJHCI* 18 (2005), 133–144.
24. Li Y., Hinckley K., Guan Z. & Landay J. A. Experimental analysis of mode switching techniques in pen-based user interfaces. In *CHI '05* (2005), 461–470.
25. Logitech 705 mouse controller. <http://www.logitech.com/en-nz/product/7108>.
26. Logitech Anywhere 905 mouse controller. <http://www.logitech.com/product/anywhere-mouse-mx>.

27. Logitech G105 gaming keyboard. <http://gaming.logitech.com/en-us/product/g105-gaming-keyboard>.
28. Apple magic trackpad. <http://www.apple.com/magictrackpad/>.
29. Malacria S., Bailly G., Harrison J., Cockburn A. & Gutwin C. Promoting hotkey use through rehearsal with exposehk. In *CHI '13* (2013), 573–582.
30. Malacria S., Scarr J., Cockburn A., Gutwin C. & Grossman T. Skillometers: Reflective widgets that motivate and help users to improve performance. In *UIST '13* (2013), 321–330.
31. McLoone H., Hinckley K. & Cutrell E. Bimanual interaction on the microsoft office keyboard. In *INTERACT'03* (2003), 49–56.
32. Microsoft Adaptive Keyboard. <http://www.microsoft.com/appliedsciences/content/projects/uist.aspx>.
33. Microsoft Touch Mouse. <http://www.microsoft.com/hardware/en-us/p/touch-mouse>.
34. Nielsen J. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1993.
35. Optimus Maximus Keyboard. <http://www.artlebedev.com/everything/optimus/maximus/>.
36. Pook S., Lecolinet E., Vaysseix G. & Barillot E. Control menus: execution and control in a single interactor. In *CHI EA '00* (2000), 263–264.
37. Putz-Anderson V. *Cumulative trauma disorders: A manual for musculoskeletal diseases of the upper limbs*. Taylor & Francis London, 1988.
38. Rekimoto J. Thumbsense: automatic input mode sensing for touchpad-based interactions. In *CHI EA '03* (2003), 852–853.
39. Ruiz J., Bunt A. & Lank E. A model of non-preferred hand mode switching. In Proc. *GI '08*, Canadian Information Processing Society (2008), 49–56.
40. Ruiz J. & Li Y. Doubleflip: a motion gesture delimiter for mobile interaction. In *CHI '11* (2011), 2717–2720.
41. Rutledge J. D. & Selker T. Force-to-motion functions for pointing. In *INTERACT '90* (1990), 701–706.
42. Saund E. & Lank E. Stylus input and editing without prior selection of mode. In *UIST '03* (2003), 213–216.
43. Tak S., Westendorp P. & van Rooij I. Satisficing and the use of keyboard shortcuts: Being good enough is enough? *Interacting with Computers* 5, 25 (2013).
44. Villar N., Izadi S., Rosenfeld D., Benko H., Helmes J., Westhues J., Hodges S., Ofek E., Butler A., Cao X. & Chen B. Mouse 2.0: multi-touch meets the mouse. In *UIST '09* (2009), 33–42.
45. Wobbrock J. O. & Myers B. A. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM ToCHI* 13, 4 (2006), 458–489.
46. Fondation X.org. <http://www.x.org/wiki/>.
47. Zhao S. & Balakrishnan R. Simple vs. compound mark hierarchical marking menus. In *UIST '04* (2004), 33–42.

Manipulation de Sliders Multiples par Franchissement

Charles Perin
INRIA, Univ. Paris-Sud,
CNRS-LIMSI
91400, Orsay, France
charles.perin@inria.fr

Pierre Dragicevic
INRIA
91400, Orsay, France
pierre.dragicevic@inria.fr

RÉSUMÉ

Le crosset est un nouvel instrument d'interaction basé sur le franchissement et exploitant la dimension orthogonale des widgets. Nous explorons l'espace de conception des crossets et illustrons leur utilisation par un cas d'application : l'exploration visuelle et le formatage de tables numériques. À notre connaissance, le franchissement n'a jamais été appliqué à la manipulation simultanée de sliders multiples ni à l'interaction avec des tables. Cet article ouvre de nombreuses perspectives et nous espérons voir plus d'interfaces basées sur cette technique dans le futur.

Mots Clés

Manipulation Directe ; Interaction Instrumentale ; Sliders ; Franchissement ; Visualisation.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous.

INTRODUCTION

Cet article présente le crosset (crossing widget), un widget franchissable pour la manipulation simultanée de sliders multiples. Par slider, nous entendons les sliders ainsi que les widgets dérivés tels que les sliders à plusieurs poignées, les boutons radio et les cases à cocher, comme nous le détaillons par la suite (Figure 4). La conception du crosset est basée sur les principes de la manipulation directe [23] et sur le modèle de l'interaction instrumentale [6, 7, 8].

Alors que les sliders sont uni-dimensionnels, des travaux ont exploité la dimension orthogonale au slider [4, 14]. Cependant, cette seconde dimension n'a jamais bénéficié du franchissement [1] (sélection de plusieurs objets par un geste unique) alors que l'interaction est adaptée à la manipulation de sliders multiples. Contrairement au pointage qui requiert de déplacer le curseur à l'intérieur d'un objet graphique et presser le bouton de la souris pour l'activer, le franchissement consiste à déplacer le curseur sur une série d'objets graphiques pour les activer.

Nous décrivons les propriétés du crosset et montrons que toute une hiérarchie de widgets peuvent hériter de ces propriétés. Nous explorons l'espace de conception des crossets, en nous basant sur une caractérisation des widgets

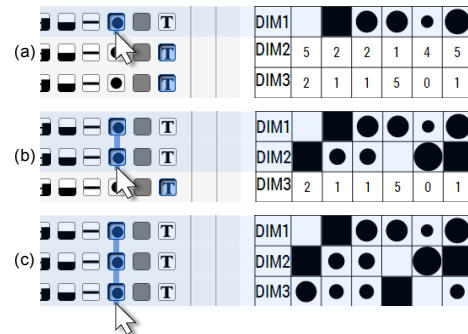


Figure 1. Illustration du franchissement de crossets radio. Les boutons permettent de sélectionner l'encodage visuel des lignes. Les lignes sont progressivement encodées par franchissement.

dérivés du slider continu standard. Nous appliquons les crossets à un projet récent [21] pour l'exploration interactive de tables numériques (Figure 1). Les crossets sont compatibles avec l'agencement des tables, génériques et polyvalents, et simples d'utilisation.

CONTEXTE

Les sliders ont des fonctionnalités multiples. Ils sont par exemple présents dans les lecteurs vidéo pour la navigation temporelle et dans les fenêtres pour faire défiler le contenu d'une page. Les sliders sont aussi souvent utilisés pour attribuer une valeur à un paramètre dans un intervalle de valeurs possibles, par exemple pour régler l'opacité d'un calque entre 0 et 100 dans un éditeur de photos.

Dimension Orthogonale

Le *degré d'intégration* [6, 7] est le rapport entre le nombre de degrés de liberté utilisés pour manipuler un instrument et le nombre de degrés de liberté de l'instrument physique (e.g., la souris) utilisé pour manipuler cet instrument. Un slider possédant un axe, il ne prend en compte qu'une dimension de l'espace et a un degré d'intégration de 1/2. Des techniques utilisent la dimension orthogonale du slider, résultant en un degré d'intégration de 2/2 [4, 14]. Cette seconde dimension est utilisée pour modifier la granularité de la navigation et augmenter la précision.

L'Infovis toolkit [14] permet de créer des sliders multi-résolution dont la précision dépend de la distance orthogonale. Certaines versions de Quicktime proposent aussi cette fonctionnalité, où la granularité de la barre de navigation temporelle est ajustée en fonction de la distance au slider.

Orthozoom [4] étend la barre de défilement classique en un instrument intégrant à la fois le pan et le zoom : déplacer la poignée le long de l'axe du slider fait défiler la page, et la dimension orthogonale est utilisée pour zoomer dans la page en fonction de la distance orthogonale au slider.

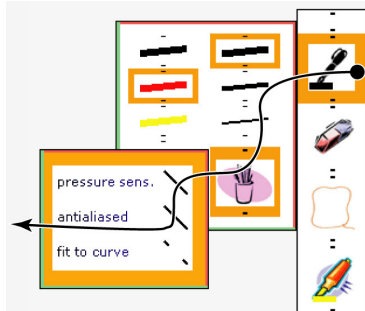


Figure 2. Exemple de geste de franchissement dans CrossY. Un geste unique permet de sélectionner successivement l'outil de dessin, le type de trait, et l'anti-aliasing.

Malgré l'existence de ces techniques, les sliders sont généralement standardisés et uni-dimensionnels. Il est en revanche fréquent de trouver des séries de sliders alignés.

Sliders Multiples

L'exemple fondateur en Infovis des sliders multiples est les requêtes dynamiques [2], pour l'exploration de données multi-dimensionnelles. Cette approche est aujourd'hui démocratisée et les requêtes dynamiques sont omniprésentes, en particulier sur les sites marchands.

Paramétrer plusieurs sliders en les modifiant un par un a du sens pour attribuer une valeur différente à chaque slider, mais est peu efficace pour donner la même valeur à plusieurs sliders. Par exemple, il est fréquent de vouloir réinitialiser l'état des sliders en leur attribuant la valeur minimale ou maximale possible. Il n'existe actuellement pas de technique permettant cette interaction alors que la dimension orthogonale du slider est inutilisée.

Franchissement

Bien que le franchissement ait été proposé en 2002 [1], très peu d'interfaces utilisent cette technique malgré plusieurs avantages par rapport au pointage, le principal étant la possibilité de "franchir" plusieurs objets d'un même geste, par exemple une série de sliders pour modifier leur valeur.

Peu de travaux sont basés sur cette technique. Dragicevic utilise le franchissement pour glisser-déposer un objet dans une fenêtre en arrière plan [13], permettant d'interagir avec un objet secondaire tout en manipulant un objet d'intérêt.

La Figure 2 illustre CrossY [3], une interface de dessin basée sur le franchissement. CrossY permet d'effectuer une sélection d'éléments dans des menus successifs en parcourant l'arborescence d'un geste continu au lieu de cliquer sur chaque élément des menus et sous-menus.

Baudisch explore les vertus du franchissement pour les interfaces contenant de nombreux boutons à deux états [5], et les Sliding Widgets [20] permettent de manipuler des widgets classiques gestuellement. En particulier, ils permettent d'interagir avec plusieurs cases à cocher simultanément par un geste unique. Cependant, ce cas reste marginal puisque les Sliding Widgets sont conçus pour manipuler un widget unique. L'auteur aborde néanmoins la manipulation simultanée de deux sliders en utilisant une aire d'activation lors du pointage, mais sans considérer la dimension orthogonale comme moyen de sélection des widgets.

Finalement, Accot et al. explorent les propriétés du franchissement [1]. Ils montrent que le franchissement effectué perpendiculairement à la forme de l'objet et de manière continue est le plus efficace. En revanche, le franchissement requiert de contrôler la direction du curseur en permanence afin de cibler les objets à franchir (voir Figure 2).

Généralisation

Les sliders alignés (disposition en grille) sont compatibles avec les tableurs (*e. g.*, Excel et Google Spreadsheet) qui permettent d'interagir avec des lignes et des colonnes (*e. g.*, sélection, tri, redimensionnement). Excel propose au moins deux manières de redimensionner des colonnes. Après sélection des colonnes à redimensionner, un clic droit ouvre un menu contextuel affichant la commande, puis une boîte de dialogue permet de spécifier au clavier la largeur des colonnes en centimètres. Le même résultat peut être obtenu en agrandissant une des colonnes sélectionnées par glisser-déposer de sa bordure. Bien que la seconde manière soit plus directe spatialement [7], le retour visuel est difficile à interpréter car l'aperçu du redimensionnement n'est disponible que pour la colonne manipulée et l'action n'est effectuée que lorsque le bouton de la souris est relâché. De plus, il est nécessaire dans les deux cas d'effectuer une séquence d'actions (sélection puis redimensionnement), impliquant une faible directitude temporelle [7].

L'interaction avec un ensemble de calques dans les logiciels tels que Photoshop illustre le besoin d'attribuer à plusieurs éléments adjacents et alignés une même commande. Photoshop permet d'ailleurs d'afficher/masquer une série de calques en maintenant le bouton de la souris pressé sur l'icône en forme d'œil et en faisant glisser la souris sur d'autres lignes de calques. Cependant, l'action n'est pas réversible et requiert toujours d'effectuer autant de tâches de pointage qu'il y a d'icônes à traverser. De plus, si le franchissement est possible pour la visibilité des calques (valeur binaire), elle n'est pas disponible par exemple pour régler l'opacité des calques (valeur continue).

Problème

Pour résumer, le slider est un widget standard largement utilisé, mais souffre des points suivants :

1. Le slider standard n'utilise qu'une dimension de l'espace alors que la dimension orthogonale est disponible.
2. Les sliders multiples sont fréquents, et manipuler plusieurs sliders individuellement peut être fastidieux.
3. L'alignement des sliders multiples n'est pas exploité.
4. Le franchissement peut apporter une solution, mais le geste associé requiert une lourde charge cognitive dû au pointage permanent en deux dimensions.

D'une part, le franchissement est optimal lorsque la trajectoire est orthogonale à l'objet à franchir, et contraindre l'interprétation du geste tout en laissant le geste de l'utilisateur libre requiert une faible charge cognitive et empêche les erreurs [1]. D'autre part, la dimension orthogonale, qui intersecte l'ensemble des sliders multiples lorsqu'ils sont disposés sous forme de grille, n'est pas utilisée. En conséquence, notre approche est basée sur le franchissement avec interprétation du geste contraint orthogonalement au widget.

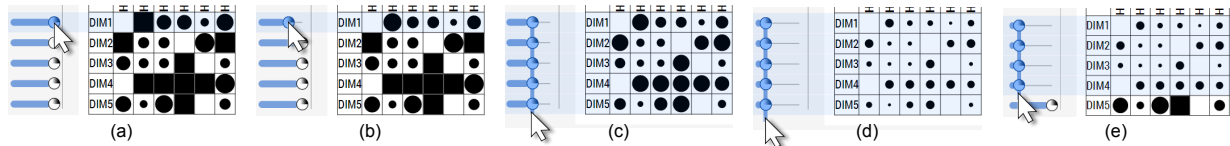


Figure 3. Franchissement de crossets (sliders continus).

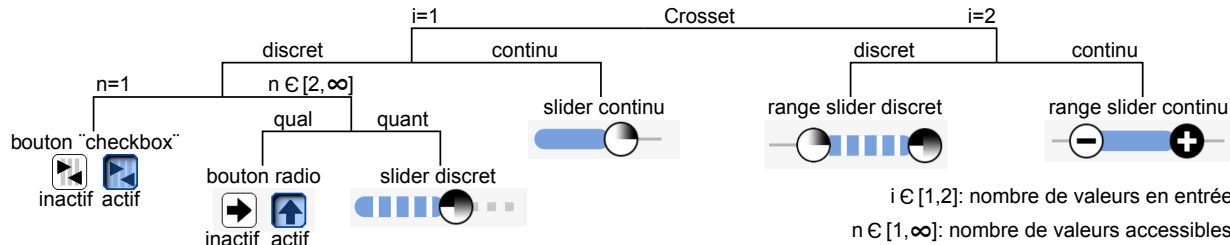


Figure 4. Les différentes implémentations de crossets.

CROSSET : WIDGET FRANCHISSABLE

Le crosset est un widget augmenté de capacités de franchissement afin de modifier simultanément des widgets adjacents. La conception des crossets détaillée ici s'applique aux interfaces en grille (tableaux, matrices, tables).

La Figure 3 illustre la technique, où les crossets (ici, des sliders continus) placés en face des lignes d'une table affectent l'encodage visuel des lignes : (a) le franchissement est activé par sélection de la poignée d'un slider ; (b) déplacer la souris horizontalement permet de naviguer dans le domaine des valeurs, comme pour un slider classique, et l'encodage visuel de la ligne associée est mis à jour ; (c) déplacer la souris verticalement permet de naviguer dans le domaine de la sélection : les crossets franchis prennent la valeur correspondant à la position horizontale de la souris ; (d) déplacer la souris horizontalement modifie la valeur des crossets franchis ; et (e) désélectionner un crosset en le franchissant à nouveau restaure sa valeur avant sélection.

La manipulation des crossets comporte trois étapes :

- Début** : le franchissement est initié par une pression du bouton de la souris sur la poignée/le bouton du crosset. La poignée/bouton change alors de représentation visuelle pour signaler à l'utilisateur sa sélection.
- Geste** : déplacer la souris dans l'espace permet de naviguer à la fois dans le domaine des valeurs et le domaine de la sélection. La représentation graphique du crosset est mise à jour afin de fournir un retour immédiat à l'utilisateur. De plus, une ligne orthogonale au slider reliant le slider au curseur indique le franchissement en cours.
- Fin** : le franchissement se termine lorsque le bouton de la souris est relâché. Des modifications peuvent s'appliquer à cette étape, comme nous le détaillons plus loin.

Types de crossets

Le crosset est générique dans le sens où il permet d'augmenter une variété de widgets standards. La Figure 4 illustre les crossets implémentés. Un crosset dépend de deux paramètres : i est le nombre de variables manipulables par l'utilisateur en entrée. Les cas les plus courants sont $i = 1$ et $i = 2$, mais Photoshop par exemple dispose de sliders à autant de poignées que désiré pour définir des gradients de couleur ; et $n \in [1, \infty]$ est le nombre de valeurs accessibles (domaine des valeurs du crosset). Puisque

le franchissement ne fait pas partie des implémentations standard, nous avons dû créer une librairie (Javascript et d3 [11]) implémentant ces widgets à partir de zéro.

- Le widget de base est le slider continu ($i = 1, n > 1$). Il se comporte comme un slider classique.
- Le slider discret est un slider continu où les valeurs possibles sont magnétisées. La représentation visuelle du crosset illustre ces valeurs discrètes.
- Le range slider continu ($i = 2, n > 1$) est un slider continu à deux poignées pour spécifier une valeur min et une valeur max. L'implémentation requiert de spécifier un écart minimal entre le min et le max afin de garantir la sélection des poignées et assurer $min < max$.
- Le range slider discret est un cas particulier du range slider continu, de la même manière que le slider discret est un cas particulier du slider continu. Les mêmes considérations s'appliquent pour l'implémentation.
- Le bouton radio est similaire au slider discret ($i = 1, n > 1$) mais est dédié aux valeurs non ordinales. Chaque bouton peut être vu comme un pas de slider discret et la sélection est exclusive (illustré Figure 1). Ce crosset ne comporte pas d'axe reliant les boutons et deux états permettent d'identifier le bouton sélectionné.
- Le bouton checkbox ($i = 1, n = 1$) peut être soit actif, soit inactif. Puisque $n = 1$, le déplacement dans le domaine des valeurs est inutile. Cependant, le domaine des valeurs pourrait déclencher des actions supplémentaires, telles que l'activation et la désactivation.

Propriétés des crossets

Le crosset respecte de nombreux principes issus de la manipulation directe et de l'interaction instrumentale.

Cohérence interne. Le crosset est un instrument générique. Il assure donc la cohérence interne de l'interface, en termes de design, propriétés graphiques, couleurs, etc. [18].

Cohérence externe. C'est la cohérence d'une interface par rapport aux interfaces existantes. Elle favorise l'apprentissage rapide et le transfert de connaissances [18, 19, 22]. Les crossets ont une grande cohérence externe puisqu'ils sont des widgets augmentés : les crossets disposent des interactions des widgets classiques. En particulier, le crosset reste compatible avec le clic (**Début** suivi immédiatement de **la Fin**, sans considérer le **Geste**). Le franchissement n'est

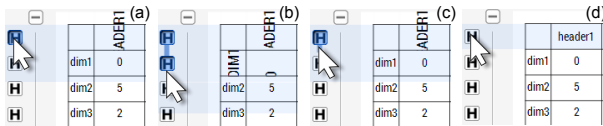


Figure 5. Après le franchissement initial, entrer à nouveau dans le premier crosset revient à annuler le geste en cours.

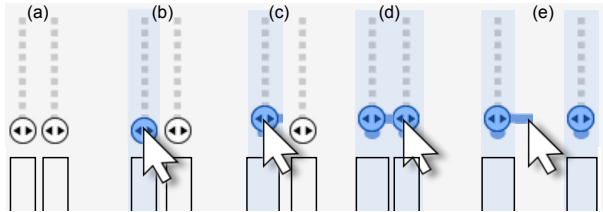


Figure 6. Problème lorsque l'action associée à un crosset modifie la position d'un ou plusieurs crossets.

enclenché que lorsque l'utilisateur en exprime le besoin en déviant orthogonalement à l'axe du widget.

Degré d'intégration. Contrairement au widget standard, qui a un degré d'intégration de 0/2 (pour les boutons) ou 1/2 (pour les sliders), les crossets ont un degré d'intégration de 2/2 puisque les deux dimensions de la souris sont utilisées.

Actions immédiates et réversibles. La représentation graphique du crosset est mise à jour afin de fournir un retour visuel immédiat [23]. De plus, le résultat est immédiat et réversible [23], comme illustré sur la Figure 3(d-e). La Figure 5 montre la réversibilité du geste pour le crosset checkbox : (a) un bouton est activé ; (b) un second bouton est activé ; (c) ce second bouton est désactivé ; (d) le premier bouton est désactivé, restaurant l'état initial.

Multi-Cibles. Le geste permet d'appliquer une action à un ensemble d'objets, ce qui est un des défis des interfaces à manipulation directe [15, 16, 17]

Décalage temporel. Puisque la sélection des objets d'intérêt et la manipulation des valeurs se fait d'un geste unique avec retour visuel immédiat, la technique minimise le décalage temporel [7] et la distance articulaire [19].

Geste contraint. Le geste de franchissement est *contraint*. Puisqu'une seule des deux dimensions est réservée au franchissement, le geste est libre mais son interprétation par le système est verrouillée dans la direction orthogonale au crosset. Cette propriété garantit que l'utilisateur ne commettra pas d'erreur de sélection [1], sans nécessiter de message intrusif interférant avec la tâche en cours [19, 22].

Actions modifiant l'agencement des crossets

Lorsque l'action associée à un crosset affecte l'agencement d'un ou plusieurs crossets, il est parfois impossible d'appliquer l'action immédiatement. Prenons l'exemple du redimensionnement de colonnes par sliders illustré sur la Figure 6 : modifier la valeur du premier crosset n'affecte pas la position du second (b-c) ; mais franchir le second crosset provoque son déplacement vers la droite (d-e), et sa désélection. Le crosset reprend alors sa valeur avant sélection et sa position initiale... et est à nouveau franchi immédiatement, puisque sous le pointeur. En résulte une succession indésirable de va-et-vient très rapides du crosset.

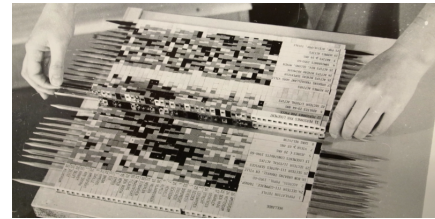


Figure 7. Manipulation d'une des matrices physiques de Bertin.

Les crossets affectant la disposition de crossets requièrent donc une stratégie différente. Notre approche est similaire au déplacement d'une fenêtre sous Windows quand les performances de l'ordinateur sont limitées, où seul le cadre de la fenêtre est affiché pendant le déplacement et la fenêtre est actualisée complètement lorsque la souris est relâchée. De la même manière, nous donnons un aperçu partiel du résultat et n'appliquons les modifications de disposition qu'au relâchement de la souris. Cela assure la compatibilité des crossets pour ce genre d'opérations, au détriment d'une réduction de la directitude temporelle de l'instrument [7].

CONCEPTION D'UNE INTERFACE BASÉE CROSSETS

Nous avons appliqué les crossets à un outil de formatage de table basé sur les matrices de Bertin [9], un dispositif physique illustré Figure 7. Bertifier [21] est une interface web reproduisant cette méthode d'analyse et exploration de données tabulaires (Figure 8, www.bertifier.fr). Cette méthode implique de nombreuses étapes, telles que le conditionnement des données, le choix de l'encodage visuel, et le réordonnancement des lignes et des colonnes.

Après avoir identifié les fonctionnalités requises, nous avons été confrontés à leur hétérogénéité. Une approche naïve aurait été de concevoir une interface WIMP proposant des boîtes de dialogue successives, comme cela a déjà été proposé [10, 12, 25]. Cependant, aucune de ces interfaces n'est parvenue à démocratiser la méthode de Bertin. Nous pensons que cela est principalement dû à la mauvaise conception des interactions, résultant en des interfaces puissantes mais particulièrement complexes à apprendre et utiliser. En revanche, certaines implémentations proposent moins de fonctionnalités mais restaurent la directitude des actions [24], s'inspirant de la méthode physique. Néanmoins, il n'existe aucune interface reproduisant cette méthode—malgré des tentatives pendant plus de 30 ans—qui soit à la fois puissante, simple et cohérente. C'est pour palier ce manque que nous avons conçu Bertifier.

Disposition des crossets

Une table est constituée de lignes et colonnes. L'objet d'intérêt peut être la cellule, comme c'est le cas pour les tableurs. Mais la méthode de Bertin spécifie que les objets d'intérêt sont les lignes et les colonnes, les actions de l'utilisateur doivent donc s'appliquer à ces objets.

Une table étant rectangulaire, il est pertinent de placer les commandes s'appliquant aux lignes et aux colonnes respectivement à gauche et au-dessus de la table. Au lieu d'adopter une démarche séquentielle 1) sélection des lignes/colonnes, 2) choix de l'action à appliquer par le biais d'un menu externe, nous dupliquons chaque action pour chaque ligne/colonne et maximisons ainsi la directitude

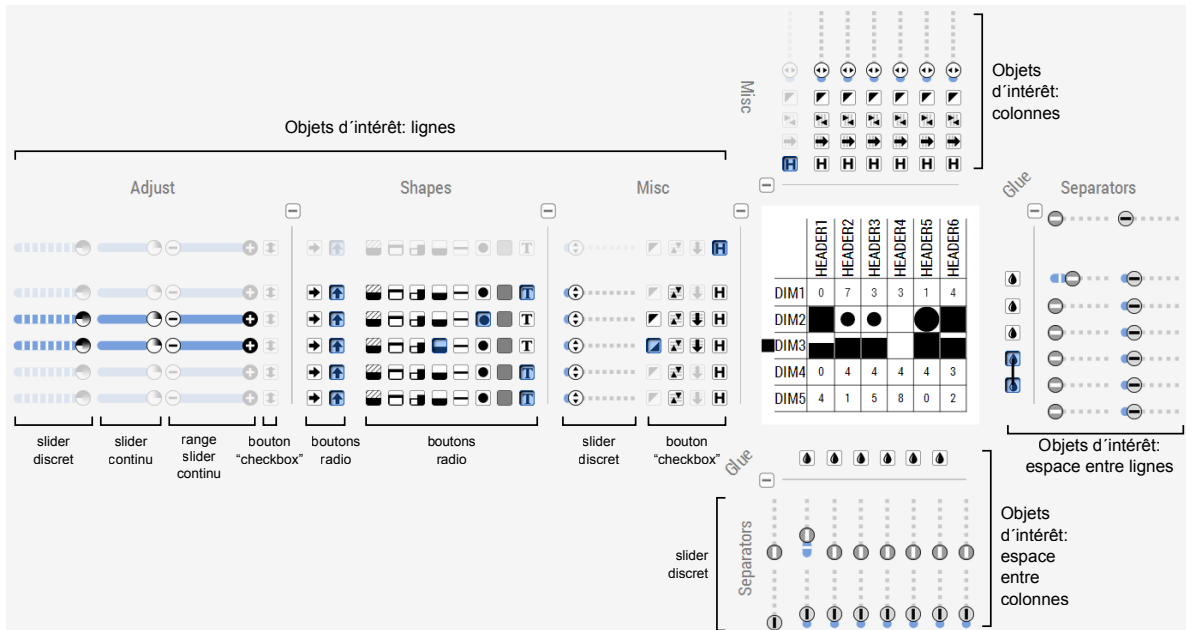


Figure 8. Interface de Bertifier. Les crosets sont disposés autour de la table.

spatiale [7]. Lors de notre étude de la méthode de Bertin, nous avons réalisé que certaines commandes ne s'appliquaient pas aux lignes/colonnes, mais aux espaces entre celles-ci ; c'est le cas des séparateurs permettant d'identifier des groupes de lignes/colonnes visuellement. Puisqu'une table a quatre côtés, nous utilisons donc l'espace à droite et en-dessous de la table pour y placer les crosets visant l'espace entre les lignes et les colonnes, respectivement. Au final, les crosets sont alignés en quatre grilles et compatibles avec le franchissement orthogonal.

Contrairement aux interactions standard des tableaux, la sélection des lignes/colonnes peut être modifiée à la volée. Afin de faciliter la correspondance entre un croset et son objet d'intérêt, nous ajoutons une surbrillance aux crosets ainsi qu'à l'objet d'intérêt cible lorsque la souris est positionnée sur un groupe de crosets. Enfin, les crosets sont regroupés en catégories expansibles afin de ne pas surcharger l'interface et de maximiser la directitude spatiale.

De plus, le positionnement des crosets autour de la table assure la visibilité et l'accessibilité des cellules. Grâce à cela, nous avons pu implémenter le glisser-déposer de lignes/colonnes et conserver la métaphore inspirée de la manipulation physique des matrices de Bertin.

Actions modifiant l'agencement spatial de la table

Nous avons évoqué les actions modifiant la disposition des crosets. Dans Bertifier, trois actions sont concernées (illustrées Figure 9) et disposent d'un retour visuel partiel.

L'encodage visuel des cellules d'une ligne/colonne en texte affecte la largeur et/ou la hauteur de la cellule. L'aperçu partiel peut consister en une rotation du texte, une écriture en lettres capitales ou minuscules, et une modification de l'ancre du texte (a). Puisque les lignes/colonnes ne sont pas modifiées immédiatement, le texte peut déborder de la cellule. Quand le geste est terminé, les lignes/colonnes sont animées vers leur nouvelle position/taille (b).

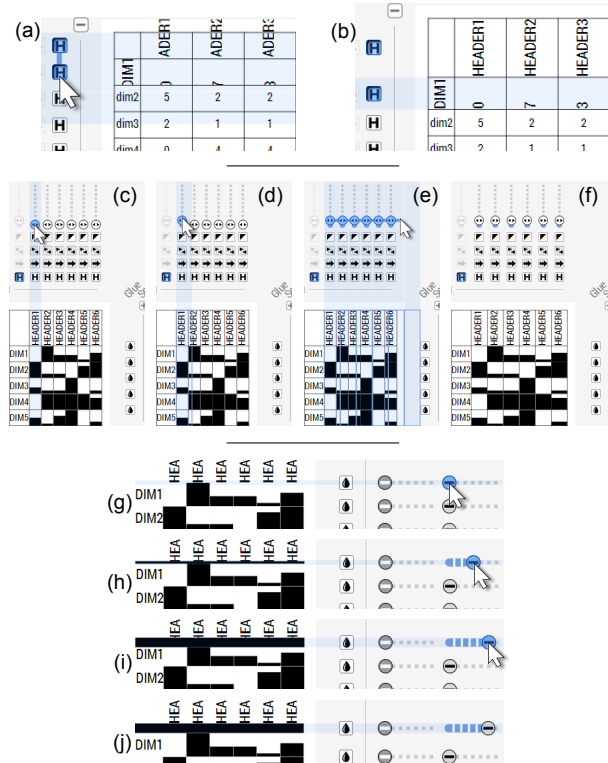


Figure 9. Trois exemples d'aperçu immédiat du résultat du franchissement où la disposition des crosets n'est mise à jour qu'au relâchement du bouton de la souris.

Le redimensionnement de lignes et colonnes affecte leur hauteur et largeur, respectivement. Dans le cas d'une colonne, après sélection d'un croset (c), déplacer verticalement la poignée change la largeur de la colonne. Nous affichons le cadre qu'auraient les colonnes si le bouton de la souris était relâché comme aperçu, afin de ne pas déplacer les crosets lors du franchissement (d-e). Lorsque le geste se termine, les colonnes sont animées vers leur nouvelle position/taille (f).

La modification de séparateurs affecte la position des lignes/colonnes. Changer la valeur d'un slider discret (g) donne un aperçu du séparateur modifié. Puisque les lignes/colonnes ne sont pas déplacées immédiatement, le séparateur s'affiche par dessus celles-ci (h-i). Comme précédemment, les lignes/colonnes sont animées vers leur nouvelle position une fois le geste terminé (j).

DISCUSSION ET LIMITATIONS

Un point crucial lors de la conception des crossets a été la décision de verrouiller la direction du franchissement perpendiculairement à l'axe du crosset. En effet, cette stratégie permet d'éviter un problème majeur des interfaces basées sur le franchissement qu'est le suivi de tunnels, une tâche moteur lente [1]. Avec les crossets, l'utilisateur peut dévier de la trajectoire "normale" sans risquer de sélectionner des crossets différents par erreur. Cela permet donc d'accélérer l'interaction en autorisant un geste approximatif sans risque d'erreur.

Limitations

Les crossets ont certaines limitations que nous discutons dans le reste de cette section.

Degré de compatibilité. Le degré de compatibilité mesure la similarité entre les actions physiques de l'utilisateur sur l'instrument et la réponse de l'instrument [7]. Les crossets ont un degré de directivité temporelle et spatiale et un degré d'intégration élevés, mais leur degré de compatibilité est discutable. Les boutons ont un degré de compatibilité élevé, mais pour certains sliders—redimensionnement des lignes et colonnes, création de séparateurs—le geste de l'utilisateur est orthogonal au résultat. Par exemple, pour augmenter/diminuer la hauteur d'une ligne, la poignée est déplacée vers la droite/gauche. Déplacer la poignée vers le haut/bas serait plus compatible avec l'action associée au slider, mais en conflit avec la dimension de sélection des crossets du geste de franchissement.

Espace Écran. Les crossets doivent être visibles à l'écran pour être franchis. Pour notre cas d'application, cela peut s'avérer problématique si la table est trop grande pour être affichée entièrement à l'écran, forçant l'utilisateur à effectuer une série de gestes couplée à un déplacement dans l'espace de travail. Cependant, le zoom résout—dans beaucoup de cas—ce problème. Une perspective prometteuse pour réduire l'espace écran nécessaire serait de rendre les crossets transients afin de n'apparaître qu'à l'activation du geste et de disparaître une fois le geste terminé.

Crossets Non-Adjacents. L'espace de conception des crossets que nous proposons s'applique à toute interface en grille, et la généralité des crossets se restreint à ces (nombreuses) interfaces : il est nécessaire que les crossets soient adjacents et les dimensions similaires. Un problème est en conséquence l'attribution d'une même valeur à des crossets non adjacents. Puisque la direction du franchissement est orthogonale aux crossets, il est impossible d'éviter un crosset. Il serait envisageable d'utiliser un modificateur clavier pour "éviter" un crosset, mais cela ajouterait de la complexité à l'interaction dont l'un des atouts majeurs est la simplicité. Nous avons contourné le problème par

les crossets discrets où le faible nombre de valeurs possibles permet d'attribuer visuellement la même valeur à des crossets éloignés. Dans le cas des tables de Bertin, le problème peut aussi être contourné en déplaçant les lignes/colonnes par glisser-déposer afin de les disposer côte à côte. Nous pensons cependant que d'autres stratégies pourraient s'avérer pertinentes, et ceci fait partie des travaux futurs que nous envisageons.

Stabilité de l'Interface. Enfin, nous avons illustré le problème de stabilité de l'interface à travers l'exemple du redimensionnement de colonnes. En particulier, le franchissement peut être perturbé lorsque l'action en cours affecte l'agencement spatial de l'interface.

Généralisation

Nous avons appliqué les crossets aux tables mais de nombreuses applications bénéficieraient de widgets franchissables. C'est le cas des requêtes dynamiques [2], où plusieurs dimensions sont manipulées. Par exemple, lors d'une recherche d'appartement, modifier simultanément la distance à A et la distance à B afin de spécifier que les deux distances ont la même importance est bénéfique ; ou, manipuler simultanément les trois sliders des couleurs dans l'espace RGB permet de spécifier un niveau de gris. L'implémentation actuelle est dédiée à des requêtes de même nature. Cependant, dans le cas fréquent où les dimensions sont différentes, attribuer la même valeur à des crossets est peu pertinent. Néanmoins, les dimensions sont souvent normalisées, permettant d'effectuer des requêtes similaires sur des sliders de sémantique différente (e. g., les indicateurs hétérogènes du Better Life Index sont exprimés entre 0 et 1). Au contraire, parfois les dimensions sont les mêmes mais leur assigner des valeurs identiques est peu souhaitable : tout dépend de la sémantique des données.

Enfin, une perspective prometteuse serait d'autoriser des gestes en 2D non contraints : effectuer une requête consisterait à dessiner une trajectoire en 2D spécifiant en un geste unique un ensemble de valeurs pour des dimensions différentes. Se pose alors la question de l'activation du mode, qui risque d'ajouter de la complexité à l'interaction.

CONCLUSION

Nous avons proposé le crosset, un nouvel instrument d'interaction franchissable pour la manipulation simultanée de widgets multiples. Nous avons détaillé l'espace de conception des crossets et proposé une série d'implémentations illustrées par un cas d'application : la méthode de Bertin [9] pour l'exploration visuelle de tables numériques.

À notre connaissance, le franchissement n'a jamais été appliqué à la manipulation de sliders multiples simultanément ni pour interagir avec des tables. Cet article ouvre des perspectives prometteuses, et nous espérons voir plus d'interfaces basées sur cette technique dans le futur.

Puisque ce travail a nécessité une implémentation à partir de zéro, nous prévoyons de créer une librairie Javascript en open source. Nous espérons que cette librairie permettra à des concepteurs de sites web, de visualisation, ou d'IHM en général, de tirer profit des crossets et d'étendre leurs fonctionnalités.

BIBLIOGRAPHIE

1. Accot J. & Zhai S. More than dotting the i's — foundations for crossing-based interfaces. In *Proc. CHI'02*, ACM (2002), 73–80.
2. Ahlberg C. & Shneiderman B. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proc. CHI '94*, ACM (1994), 313–317.
3. Apitz G. & Guimbretière F. Crossy: A crossing-based drawing application. In *Proc. UIST'04*, ACM (2004), 3–12.
4. Appert C. & Fekete J.-D. Orthozoom scroller: 1d multi-scale navigation. In *Proc. CHI '06*, ACM (2006), 21–30.
5. Baudisch P. Don't click, paint! using toggle maps to manipulate sets of toggle switches. In *Proc. UIST '98*, ACM (1998), 65–66.
6. Beaudouin-Lafon M. Interaction instrumentale: de la manipulation directe à la réalité augmentée. In *Proc. IHM '97* (1997).
7. Beaudouin-Lafon M. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proc. CHI '00*, ACM (2000), 446–453.
8. Beaudouin-Lafon M. & Mackay W. E. Reification, polymorphism and reuse: Three principles for designing visual interfaces. In *Proc. AVI '00*, ACM (2000), 102–109.
9. Bertin J. *La graphique et le traitement graphique de l'information*. Nouvelle bibliothèque scientifique. Flammarion, 1975.
10. Bertin J. & Chauchat J.-H. Logiciel amado (analyse graphique d'une matrice de données). *CISIA, France* (1994).
11. Bostock M., Ogievetsky V. & Heer J. D3 data-driven documents. *IEEE TVCG 17*, 12 (2011), 2301–2309.
12. Caraux G. & Pinloche S. Permutmatrix: a graphical environment to arrange gene expression profiles in optimal linear order. *Bioinformatics 21*, 7 (2005), 1280–1281.
13. Dragicevic P. Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. In *Proc. UIST '04*, ACM (2004), 193–196.
14. Fekete J.-D. The infovis toolkit. In *Proc. INFOVIS '04*, IEEE (2004), 167–174.
15. Frohlich D. M. The history and future of direct manipulation. *Behaviour & Information Technology 12*, 6 (1993), 315–329.
16. Frohlich D. M. Direct manipulation and other lessons. In *Handbook of human-computer interaction (2nd ed)*, Elsevier (1997), 463–488.
17. Gentner D. & Nielsen J. The anti-mac interface. *Commun. ACM 39*, 8 (1996), 70–82.
18. Grudin J. The case against user interface consistency. *Commun. ACM 32*, 10 (1989), 1164–1173.
19. Hutchins E. L., Hollan J. D. & Norman D. A. Direct manipulation interfaces. *Hum.-Comput. Interact. 1*, 4 (1985), 311–338.
20. Moscovich T. Contact area interaction with sliding widgets. In *Proc. UIST '09*, ACM (2009), 13–22.
21. Perin C., Dragicevic P. & Fekete J.-D. Revisiting bertin matrices: New interactions for crafting tabular visualizations. *IEEE TVCG* (2014).
22. Shneiderman B. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology 1*, 3 (1982), 237–256.
23. Shneiderman B. Direct manipulation: A step beyond programming languages. *Computer 16*, 8 (1983), 57–69.
24. Siirtola H. Interaction with the reorderable matrix. In *IV*, IEEE (1999), 272–279.
25. Wu H.-M., Tien Y.-J. & Chen C.-h. Gap: A graphical environment for matrix visualization and cluster analysis. *Comput. Stat. Data Anal. 54*, 3 (2010), 767–778.

The Hotkey Palette: Flexible Contextual Retrieval of Chosen Documents and Windows

Jonathan Aceituno

Inria Lille – Nord Europe, Université Lille 1
59650, Villeneuve d'Ascq, France
jonathan.aceituno@inria.fr

Nicolas Roussel

Inria Lille – Nord Europe
59650, Villeneuve d'Ascq, France
nicolas.roussel@inria.fr

ABSTRACT

We present the Hotkey Palette, a quasi-modal interface enabling quick retrieval of chosen documents and windows by defining and triggering keyboard shortcuts either on the physical keyboard or with an on-screen keyboard. The Hotkey Palette improves on previous work by providing flexible contextualization of shortcuts that leverages document hierarchies, and by merging document and window retrieval in a single interface. The paper describes the design and implementation of the interface and presents novel use cases for document and window management.

Key Words

on-screen keyboard; document retrieval; interface customization; window management

ACM Classification Keywords

H.5.2 Information interfaces and presentation (e.g. HCI): User interfaces - Graphical user interfaces.

INTRODUCTION

Users of Personal Computers interact with a large number of resources¹ to do their work. To handle their different tasks, they need their documents to be readily available, and as the number of activities and documents increase, systems must offer adequate support for quick retrieval of these resources.

Document retrieval is primarily done by using a file browser, but navigation in a folder hierarchy takes a long time [3]. The desktop, the Windows Start Menu, OS X's Dock and similar *quick retrieval facilities* help decrease retrieval times by providing customizable areas on which shortcuts to chosen resources or the resources themselves can be placed. They are always available and thereby useful to keep things close at hand. But there is no way to indicate which resource is useful for which task, and the more things there are, the more time it takes to find the documents relevant to the task at hand in these places.

1. In this paper, the resources or documents we consider are local files, folders and applications, but also remote web pages, and, more generally, anything that can be identified with a uniform resource locator (URL) [4].

Quick access to specific resources can also be provided by keyboard shortcuts. But the facilities provided to define these shortcuts are usually hard to use, and remembering them all is difficult. Graphical representations reifying the shortcuts and supporting direct manipulation can help. Touch-display keyboards (TDK [5]) would make it possible to create a keyboard shortcut by dragging a document to the desired key, for example. But such keyboards are currently uncommon and such a simple approach does not scale: one not only needs shortcuts but also ways to organize them. Existing on-screen keyboards, such as *Qliner hotkeys*², share the same scaling issues and do not convey the state of the resource.

The way people configure quick retrieval facilities depends on the context in which resources will be used [17]. Some resources pertain to specific projects while others are more general, and the users' workarounds to put resources close to where they might need them, as described in [17], suggest a lack of support for resource contextualization in current retrieval facilities. When all the needed documents for a task are found and open, the problem furthermore shifts from document retrieval to window management. Unlike document retrieval facilities, many window managers support the notion of task context by proposing window grouping [12] or virtual desktops [7]. However, window managers usually do not take the user's previous efforts on resource organization into consideration and require additional management that could be avoided.

Recent systems [2, 16, 8] help solve the contextualization problem by explicitly representing the activities used to organize resources and work around them. Activity-centric environments help effectively manage access to related windows *and* documents using the same notion of activity. For example, Giornata [16] not only maps windows of different activities to different virtual desktops, but also associates the files on each desktop to the relevant activity, which is particularly appreciated by users. Yet activities have blurry boundaries and users have trouble defining and separating them [1]. Moreover, activity-based approaches do not support finer-grained (linking a document to another) or coarser-grained (associating a document to several activities) contextualizations.

In this paper, we present the Hotkey Palette (Figure 1), a quick retrieval facility that uses hotkeys and makes them visible and configurable through a quasi-modal [10] always-available on-screen keyboard. This facility contributes to the state of the art in three ways:

2. <http://www.qliner.com/hotkeys>



Figure 1. Hotkey Palette displayed on the bottom of the display, on top of other windows.

- it extends on-screen keyboard interaction by providing feedback on the state of the linked resources;
- it provides persistent and integrated access to local windows and files and other online resources;
- it provides flexible control over contextualization by leveraging existing resource hierarchies.

The rest of this paper is organized as follows. We first describe the design, the interaction techniques and the implementation of the Hotkey Palette. We then describe different usage patterns and finally conclude with a description of future work.

THE HOTKEY PALETTE

The Hotkey Palette (Figure 1) uses an on-screen keyboard to visualize and configure persistent shortcuts to access chosen documents and windows. It appears without delay when the user holds a particular modifier key, so as to provide immediate visual information to help key selection without hindering expert performance. In our implementation, we chose the *F_n* key of Apple keyboards, to avoid interfering with system shortcuts. The Hotkey Palette allows to access documents and windows by pressing associated keys while holding the modifier, and disappears when it is released. In this section, we explain how contextualization works, describe design of the Hotkey Palette and the interaction techniques it affords before detailing its current implementation.

Supporting contextualization

The following example describes our approach to the problem of resource contextualization that was described in the introduction, using the notions of *contextual shortcuts* and *hierarchical specialization* that we will detail later on.

Imagine a user, Alice, filing her projects on a local hard disk using the following hierarchy: the `Documents` folder contains two folders, `Work` and `Personal`, that each contain folders for her work and personal projects respectively. She already has a *global* hotkey providing a shortcut to her `Documents` folder from anywhere using *F_n+D*. But she has two different to-do lists, for work and personal projects, and wants to have them handy. She creates two *contextual* shortcuts mapped on *F_n+T*: one in her work folder mapping it to her work to-do list, and

the other in her personal folder mapping it to the personal one. Thanks to *hierarchical specialization*, pressing *F_n+T* while working on a work project (i.e. any document in `Work` or its sub-folders) will thus open the work to-do list, but pressing the same *F_n+T* while working on a personal project (i.e. any document in `Personal` or its sub-folders) will open the personal to-do list.

One of Alice’s personal projects is to maintain the web site of the local poker league, and she frequently receives e-mails from members prompting her to announce tournaments on this site. To support her workflow, she has created a contextual shortcut to the local copy of the web site’s announcements page associated to her mail application, and another to a synchronization script associated to the entire web site’s local copy (i.e. the enclosing folder). When she receives an announcement mail, she uses the former shortcut to quickly access the announcements page and modify it. From this document, through *hierarchical specialization*, she can use the latter to synchronize the online web site with the modified local copy.

We can see in this example that contextual shortcuts differ from global ones in that they are associated to a specific resource (interaction details are discussed later). They are only available when that resource is the current object of interest (the one displayed in the active window), thus providing context-specific access to others. *Hierarchical specialization* makes it possible to support the same contextual shortcuts on different related resources by exploiting their hierarchical relation.

In the above example, several shortcuts are mapped to the same key. To determine which shortcut is the most relevant to the current resource of interest, we use the following relevance order:

- contextual shortcuts pertaining to this object (*X*),
- contextual shortcuts pertaining to any resource *Y* such that *Y* is an ancestor of *X*³ (the closer, the better),
- global shortcuts.

3. Every document has a URL which path can be used to determine the relative location to another, if they live in the same hierarchy. The folder `file:///Documents` is an ancestor of the file `file:///Documents/Work/IHM14 Paper.doc` for example.

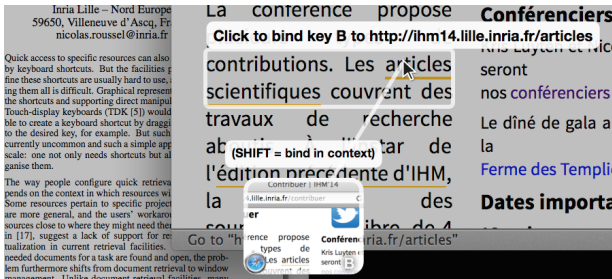


Figure 2. Holding $Fn+B$ and pointing to a hyperlink to configure a shortcut to it on key B.

Hierarchical specialization is the outcome of the above rule: when a contextual shortcut is associated to a folder, it will be available by default for all its descendants.

Interaction

In the example above, the user created and triggered shortcuts. Here, we detail the interaction techniques that support these functionalities and more advanced ones.

Configuring a hotkey

Pressing Fn and some unused key X creates a global shortcut mapping $Fn+X$ to the current resource of interest. Holding $Fn+X$ while clicking on an on-screen resource creates a global shortcut to it (Figure 2, left). Holding $Fn+Shift+X$ when clicking instead results in a contextual shortcut to it associated to the current resource of interest (Figure 2, right). When the user is holding $Fn+X$ or $Fn+Shift+X$, the palette disappears to prevent occluding potential targets, and the current resource of interest and the resource under the mouse pointer are highlighted.

Triggering a hotkey

To trigger an existing shortcut mapped on a particular key, e.g. X , the user must hold the Fn modifier and press X . This brings the associated resource to the foreground. As an alternative, the user can click on the desired key when the Hotkey Palette is shown. Like command hotkeys whose meaning can change according to the active application, the resource associated with a key may depend on the current resource of interest. Since shortcuts are persistent, the triggered one may be associated to a resource that is not available on-screen anymore. In this case, the resource is re-opened, if possible. If not, a system beep is issued.

Visual design

The Hotkey Palette appears on the bottom of the user's main display as a semi-transparent on-screen keyboard sized to fill the whole horizontal space (Figure 1). Only the keys on the three letter rows of the keyboard⁴ are used. This way, each object associated to a key is displayed in a square taking up about $1/11$ of the horizontal resolution of the main display (174 pixels on a 1920x1080 display, 93 pixels on a 1024x768 display).

What is displayed on each key of the palette depends on the resource associated to it. If there is no shortcut, the key is just displayed with its character as an empty square

4. In a standard QWERTY keyboard, these rows correspond to characters Q to P, A to ; , and Z to / .

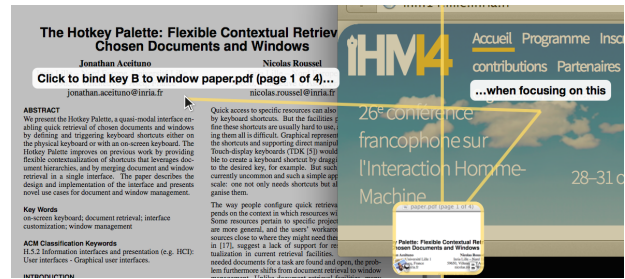


Figure 3. Holding $Fn+Shift+P$ and pointing to a document to configure a contextual shortcut to it associated to the front window on key P.

with a white border (Figure 4a). If the key is associated to a local or remote document (e.g. a web page), an up-to-date preview is displayed together with its title, filling the whole square (Figure 4b). If a preview is not available, its title and/or URL is displayed instead (Figure 4c). If the key is associated to a window, its aspect ratio is modified to fit the window's, and an up-to-date and unscaled preview of the window centered on its title is displayed together with the application icon (Figure 4d). We chose this presentation because we found it easier to distinguish between similar windows than with scaled-down screenshots.

If a window that does not exist anymore can be re-opened, its preview is yellowed like an old photograph to indicate that it is not up-to-date (Figure 4e). If the window or resource cannot be found anymore, for example because it has been deleted, its preview is grayed to indicate that accessing the resource is no longer possible (Figure 4f).

The border of a key indicates the level of contextuality of its shortcut. A white border is used if the shortcut is global. A yellow border (Figure 4g) indicates that the shortcut is associated to the currently active resource. Finally, a red border (Figure 4h) indicates that it is associated to one of its hierarchical ancestors.

Graphical interaction

When the Hotkey Palette is visible, advanced graphical interaction with the shortcuts is possible. They can be moved around the keyboard using drag-and-drop. Users can drag a document from any application, hold Fn and drop it on a key to create a global shortcut (as before, holding $Fn+Shift$ results in a contextual shortcut related to the current resource of interest). Dropping a document on a key containing an application or a script will launch the script or application using the dropped document as a parameter. Dragging a shortcut to a document from a key and dropping it in an external window will have the same effect as dropping the document itself. The palette occupies a large part of the screen, but it can be hidden at any time by releasing the modifier key, even in the course of a drag-and-drop action. A right click on a key shows a menu with options to delete the shortcut and, for contextual ones, to bring up the resource defining the context.

Implementation

Our prototype is implemented in Objective-C with Cocoa and runs under OS X. We used the Quartz Window Services API to retrieve window information, and the Acces-



Figure 4. Closeup on the top left keys of the Hotkey Palette above the active window: (a) empty key; (b) document preview; (c) document title when the preview is not available; (d) window preview; (e) yellowed preview of a closed window; (f) grayed preview of a deleted document; (g) contextual shortcut associated to the active window (yellow border); (h) contextual shortcut associated to an ancestor of the document shown in the active window.

sibility API to inspect objects under the mouse, as well as to get specific information on the contents of a window, such as the document it represents. We also used the bookmarks API to make shortcuts resilient to renamed or moved documents, ensuring that they can always be opened, and the previews of local documents and remote web pages are generated using the Quick Look API and WebKit, respectively.

To ensure persistence over time, for example to be able to re-open a window that has long been closed, the program saves specific cues about the corresponding resource and its context whenever a shortcut is created. These cues are different according to the type of the resource: a simple URL for remote resources, and a relocation-resilient bookmark for local resources. For windows, cues include the application identifier, the window identifier in the accessibility API, its title, and a relocation-resilient bookmark for the document it represents, if any. When the program loses track of a window, it tries to match the saved cues to each of the opened ones to find it again. If it cannot be found, the program tries to locate the resource that was displayed in the window and to re-open it.

USAGE PATTERNS

In this section, we present some of the usage patterns that the distinctive features of the Hotkey Palette enable, namely: reminding the user of something to do, monitoring changes, improving web page revisitation, and cycling between specific documents or windows.

Reminders and monitors

Shortcuts to resources can be used as reminders of related actions to execute in the future. These reminders can be contextual shortcuts, so that the user chooses *when* to be reminded of them. For example, Alice may create a contextual shortcut associating a photograph of her coworker Bob to her `Work` folder so that the next time she works on a project in this folder and brings up the Hotkey Palette, she will incidentally be reminded of inviting him to the next poker tournament.

Because it shows large and up-to-date previews, the Hotkey Palette can also be used to monitor changes on

documents and web pages. Although the current implementation does not indicate whether the document has changed since the last visit, users can still benefit from the up-to-date preview to check, for example, if Alice has indeed updated the poker league's website as expected.

Alternative bookmarks

Web page revisitation is primarily supported in web browsers by bookmarks and history. The Hotkey Palette can also act as an alternative bookmark facility, or an extended version of PageLinker [15]: here, one can not only associate a web page to another, but contextual links can also be created between web pages and any local resource, in both directions. Like with PageLinker, the query part of the URL is not taken into account for context to avoid tying shortcuts to a particular web session. Hierarchical specialization works in this case because web URLs have paths.

Local cycling

In his description of the *memex* [6], Vannevar Bush defined associative trails as a chain of documents linked together, sometimes branching off to side trails. The imagined interactions with these trails involved only hitting a few keys. While the Hotkey Palette is not designed to handle trail navigation, this principle can still be useful to quickly travel along a trail of related resources by creating contextual shortcuts mapped to the same key and associating these resources.

Alice could use this approach to create a personal web-bring linking her preferred poker news sites, for example. But she could also use it to cycle between a specific list of applications. She could use the *C* key to cycle between all her communication applications, for example. To do so, she would first create a global shortcut mapping that key to the contact list of her instant messaging application. She would then define a contextual shortcut from this window to her mail application, using the same key. She would continue with the websites or applications of her various social networks. Pressing *F_n+C* on any of these resources would then bring up the next one, the global shortcut to the instant messaging application closing the ring.

DISCUSSION

Extensive use of the Hotkey Palette during its development allowed us to refine the design and witness the emergent usage patterns enabled by contextual shortcuts and hierarchical specialization that are described in the previous sections. A longitudinal study would bring deeper understanding of the benefits and drawbacks of these ideas in document and window management.

The focus of this paper was more on introducing and demonstrating these concepts, by way of the Palette, than in validating interface design choices. The latter is required before deployment to real users but might not be trivial. For example, choosing *F_n* as the modifier is convenient for our demonstration, but is not adapted for real-world use because it is located on the bottom left of the MacBook keyboard, thus making key combinations like *F_n+P* difficult to reach with one hand.

Nevertheless, contextual shortcuts and hierarchical specialization being intended to enable *quick retrieval*, we strived for a minimal amount of effort in order not to discourage the use of the Hotkey Palette. Further study will ultimately confirm whether our design was successful in this regard. It would also help determine how to best incorporate the Hotkey Palette into existing tasks and encourage its use where beneficial.

CONCLUSION

In this paper, we presented the Hotkey Palette, a quick retrieval facility based on user-defined shortcuts with the following distinctive features: quasi-modal interaction with an on-screen keyboard, persistent and integrated access to local windows and files and other online resources, and flexible contextualization based on resource hierarchies. We detailed the peculiarities of our design, the associated interaction techniques and our current implementation. Finally, we showed how the technique allows novel uses in window and document management.

By merging two input modalities, pointing and typing, on a shared representation, on-screen keyboards may provide benefits for intermodal learning [14], as syntactically similar actions (e.g. pressing a key vs. clicking an on-screen key) have the same effects. In addition, they provide a spatially stable layout, allowing users to rely on their special memory, which could lead to faster retrieval times [11, 13]. Both of these hypotheses require further investigation. In the future, we would also like to investigate how the design of the Hotkey Palette could be used to improve learning of application hotkeys, and how it compares to recent work [9].

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their helpful feedback and insightful suggestions.

REFERENCES

- Bardram J. E. *Beyond the Desktop Metaphor: Designing Integrated Digital Work Environments*. MIT Press, 2007, ch. From Desktop Task Management to Ubiquitous Activity-Based Computing, 224–259.
- Bardram J. E., Bunde-Pedersen J. & Soegaard M. Support for activity-based computing in a personal computing operating system. In Proc. *CHI '06*, ACM (2006), 211–220.
- Bergman O., Whittaker S., Sanderson M., Nachmias R. & Ramamoorthy A. How do we find personal files?: The Effect of OS, Presentation and Depth on File Navigation. In Proc. *CHI '12*, ACM (2012), 2977–2980.
- Berners-Lee T., Fielding R. & Masinter L. Uniform resource identifier (URI): Generic syntax. RFC 3986, 2005.
- Block F., Gellersen H. & Villar N. Touch-display keyboards: Transforming keyboards into interactive surfaces. In Proc. *CHI '10*, ACM (2010), 1145–1154.
- Bush V. As we may think. *Atlantic Monthly* 176 (1945), 101–108.
- Henderson Jr. D. A. & Card S. Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Trans. Graph.* 5, 3 (1986), 211–243.
- Houben S., Vermeulen J., Luyten K. & Coninx K. Co-activity manager: Integrating activity-based collaboration into the desktop interface. In Proc. *AVI '12*, ACM (2012), 398–401.
- Malacria S., Bailly G., Harrison J., Cockburn A. & Gutwin C. Promoting hotkey use through rehearsal with *exposehk*. In Proc. *CHI '13* (2013).
- Raskin J. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., 2000.
- Robertson G., Czerwinski M., Larson K., Robbins D. C., Thiel D. & van Dantich M. Data mountain: Using spatial memory for document management. In Proc. *UIST '98*, ACM (1998), 153–162.
- Robertson G., Horvitz E., Czerwinski M., Baudisch P., Hutchings D. R., Meyers B., Robbins D. & Smith G. Scalable fabric: Flexible task management. In Proc. *AVI '04*, ACM (2004), 85–89.
- Scarr J., Cockburn A., Gutwin C. & Bunt A. Improving command selection with *commandmaps*. In Proc. *CHI '12*, ACM (2012), 257–266.
- Scarr J., Cockburn A., Gutwin C. & Quinn P. Dips and ceilings: Understanding and supporting transitions to expertise in user interfaces. In Proc. *CHI '11*, ACM (2011), 2741–2750.
- Tabard A., Mackay W., Roussel N. & Letondal C. *Pagelinker*: Integrating contextual bookmarks within a browser. In Proc. *CHI '07*, ACM (2007), 337–346.
- Voida S., Mynatt E. D. & Edwards W. K. Re-framing the desktop interface around the activities of knowledge work. In Proc. *UIST '08*, ACM (2008), 211–220.
- Zacchi A. & Shipman F. Personal environment management. In *Research and Advanced Technology for Digital Libraries*, L. Kovács, N. Fuhr, and C. Meghini, Eds., vol. 4675 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, 345–356.

L'accident du vol AF447 Rio-Paris, un cas d'étude pour la recherche en IHM

Stéphane Conversy

Stéphane Chatty

Hélène Gaspard-Boulinec

Jean-Luc Vinot

Université de Toulouse - ENAC
7 av. Edouard Belin, 31055 Toulouse, France
prenom.nom@enac.fr

RÉSUMÉ

Le 1er Juin 2009, le vol AF447 de Rio à Paris s'abîmait dans l'Océan Atlantique. Les expertises de sécurité et judiciaires ont conclu à un rôle important des facteurs humains dans cet accident. Observant que de nombreux éléments du rapport d'enquête produit par le Bureau d'Enquête et d'Analyse du Ministère des Transports français peuvent être assimilés à des concepts connus de l'IHM, nous proposons d'en faire un cas d'étude pour la recherche en IHM. Après avoir introduit le vocabulaire aéronautique nécessaire à leur compréhension, nous reprenons les éléments du rapport en les assimilant, organisant puis traduisant dans des cadres conceptuels du modèle de l'action et de l'activité scientifique. Nous espérons ainsi favoriser des travaux de recherche visant à une modélisation plus formelle de l'accident, ou encore à l'identification d'améliorations possibles dans les systèmes de bord.

Mots-clés : accident, aéronautique, théories de l'IHM

ACM Classification Keywords : H.5.2 Information Interfaces and presentation : User Interfaces

INTRODUCTION

L'appareil Airbus A330 du vol Air France AF447 du 1er Juin 2009 entre Rio et Paris s'est abîmé dans l'océan Atlantique, entraînant le décès des 228 personnes à son bord. Le Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile (BEA) appartenant au Ministère des Transports Français a produit un rapport suite à une enquête de sécurité sur les circonstances de l'accident [2]. Comme le BEA le stipule, « ses enquêtes ont pour unique objectif l'amélioration de la sécurité aérienne et ne visent nullement la détermination des fautes ou responsabilités ». Le rapport vise néanmoins à analyser les causes et l'enchaînement des conséquences qui ont conduit à l'accident.

Comme souvent dans de tels cas, les causes peuvent être nombreuses. L'aviation est un système socio-technique complexe composé d'acteurs multiples (agences nationales et internationales, constructeurs, compagnies, organismes de formation, pilotes etc.). Le rapport d'enquête

comporte des sections concernant le déroulement du vol, les pilotes et leur comportement, le matériel, les conditions météorologiques. L'enquête s'appuie notamment sur les enregistrements effectués à bord dans ce qu'on appelle communément les « boîtes noires ».

Le rapport semble exclure toute panne des systèmes embarqués, à l'exception des capteurs appelés « sondes Pitot ». La défaillance de ces sondes est un cas prévu : les systèmes de bord sont conçus pour y réagir, et ont eu le fonctionnement attendu par leurs concepteurs. C'est donc une combinaison des comportements de ces systèmes, des conditions de vol et des réactions des pilotes, qui a donné une issue fatale à cette défaillance. Cela fait de cet accident un cas d'étude particulièrement pertinent pour la recherche en interaction homme-machine, d'autant que le rapport fournit des éléments très détaillés.

Le rapport offre des analyses sur les fonctionnements de l'interface homme-machine du cockpit, sur les raisonnements que les pilotes ont pu tenir devant les informations données par le système, et sur les interactions entre sous-système humains et matériels. Cette analyse se base sur les échanges verbaux, l'enregistrement des actions effectuées par les pilotes et des réponses des sous-systèmes, et des simulations effectuées pour vérifier notamment le comportement des différentes sorties visuelles, auditives ou haptiques des sous-systèmes matériels. Pour qui comprend le vocabulaire aéronautique, cet ensemble de données est très riche et rappelle en de nombreux points des questions traitées par la recherche en IHM.

Le présent article vise à rendre accessible ce cas d'étude à la communauté académique de l'IHM. Le but est de servir : à la formation des concepteurs de systèmes, à tester que des modèles théoriques rendent compte fidèlement des phénomènes décrits, ou encore à la recherche sur la formation des pilotes ou sur la conception des systèmes embarqués. Pour cela, nous commençons par introduire le vocabulaire aéronautique nécessaire. Nous reprenons ensuite les extraits de texte du rapport du BEA qui nous semblent les plus pertinents pour l'IHM. Nous reproduisons d'abord les conclusions du rapport liées à l'interaction homme-machine et aux facteurs humains. Puis nous sélectionnons cinq phénomènes mentionnés dans le rapport. Pour chacun d'entre eux, nous montrons comment il peut être relié aux cadres d'analyse de l'IHM et de l'ergonomie, en particulier le modèle de l'action de Norman [15], ou de l'épistémologie, en particulier l'abduction [11].

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ème} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670459>

LE DOMAINE DE VOL D'UN AVION ET SON PILOTAGE

Le rapport du BEA décrit l'accident du vol AF447 comme une sortie de l'avion de son domaine de vol, suite à un mauvais fonctionnement temporaire de capteurs de pression et aux réactions de l'équipage que cela a engendrées. Nous introduisons ci-dessous les éléments nécessaires pour comprendre les extraits du rapport.

Vol et domaine de vol

Pour qu'un avion vole, il faut que l'écoulement aérodynamique produise une force vers le haut suffisante par rapport à son poids. Cette force de « portance » dépend de nombreux paramètres : vitesse par rapport à l'air, angle d'attaque du flux d'air par rapport aux ailes (incidence), configuration des ailes (volets rentrés ou sortis), altitude.

L'avion ne vole en sécurité que s'il reste dans son domaine de vol, défini par une combinaison des domaines de valeur des différents paramètres. Ainsi, à l'instar de la vitesse du son, les vitesses compatibles avec le domaine de vol dépendent de la température de l'air et donc de l'altitude : l'unité de vitesse en nombre de Mach (rapport à la vitesse du son) est donc plus pertinente qu'en noeuds pour raisonner sur le domaine de vol à haute altitude. Si l'incidence est trop grande ou la vitesse est trop faible, la masse d'air ne porte plus l'avion ; on dit qu'il décroche. On sort d'un décrochage en donnant à l'avion une « assiette à piquer », nez vers le bas, réduisant ainsi l'incidence. La valeur maximum d'incidence diminue lorsque le nombre de Mach augmente, ce qui signifie qu'à Mach élevé (généralement à haute altitude), l'avion entre en « décrochage haut » à un angle d'incidence inférieur à celui à vitesse faible. À haute vitesse, des phénomènes aérodynamiques provoquent une perte de portance et des vibrations dangereuses pour l'avion (« survitesse »). Ces phénomènes n'ont cependant pas été rencontrés sur les A330 [2].

La mesure des paramètres est effectuée par des capteurs physiques : les sondes Pitot mesurent la pression d'air totale dont on soustrait la pression statique pour obtenir la pression produite par la vitesse de l'avion et une mesure de cette vitesse ; des gyroscopes mesurent l'assiette et l'inclinaison (à gauche ou à droite) ; des sondes mesurent l'incidence. Les avions modernes comme l'A330 sont pilotés par des sous-systèmes humains et automatiques en interaction [1]. Ce système hybride a notamment pour rôle de faire en sorte que l'avion reste dans son domaine de vol.

Pilotage

Deux pilotes sont aux commandes (fig. 1[1]), le pilote en fonction (PF, à droite dans le cockpit lors de l'accident) et le pilote non en fonction (PNF, à gauche dans le cockpit). Chacun dispose de dispositifs d'entrée, notamment d'un sidestick (joystick isométrique latéral) revenant à sa position initiale quand il est relâché. Les mouvements longitudinaux sur le sidestick ont une influence sur l'assiette de l'avion, par des actions à « cabrer » (tirer pour faire monter le nez de l'avion) et à « piquer ». Les mouvements latéraux ont une influence sur l'inclinaison.

Le dispositif de sortie consiste en un ensemble d'écrans, de voyants et de haut-parleurs et ceux qui concernent l'accident sont :

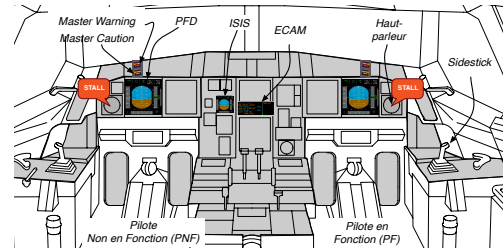


Figure 1. Le cockpit et ses éléments [1]

- l'écran de vol primaire (PFD), sur lequel sont affichées les informations considérées comme primordiales : vitesse, horizon artificiel, altitude, cap (fig. 2) ; et l'ISIS, un système de secours reprenant ces informations ;
- l'écran de supervision des systèmes (ECAM), sur lequel sont affichées des notifications produites par les sous-systèmes de mesure (fig. 3) et les voyants Master Warning (rouge) et Master Caution (orange) qui signalent des notifications de l'ECAM de niveau respectif ;
- l'alarme de décrochage, qui prévient de façon visuelle (Master Warning qui clignote) et sonore (séquence en boucle de quatre « buzz » de quelques dixièmes de seconde (« cricket »), et message vocal « STALL » (« décrochage »)) que l'avion est hors du domaine de vol.

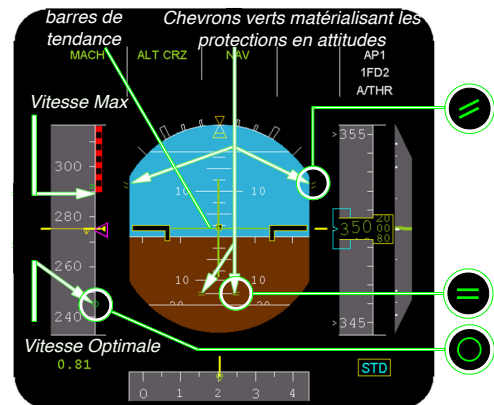


Figure 2. PFD en loi Normale (d.)

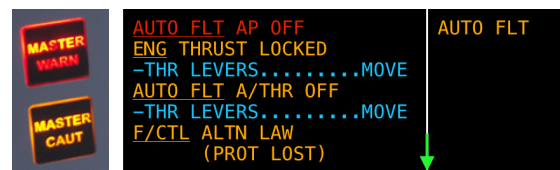


Figure 3. Master Warning & Caution (g.) et ECAM (d.)

Les instruments et appareil de navigation permettent aux pilotes de « voler aux instruments » dans les espaces aériens où cela est obligatoire, c'est-à-dire de voler sans condition de visibilité. Cependant, les pilotes peuvent aussi utiliser des informations de contexte : signaux visuels par le vitrage (p. ex. feux de saint elme), ressenti des vibrations de l'appareil (*buffet*, notamment en approche de décrochage), sons extérieurs (bruit aérodynamique, cristaux de glace), voire odeur d'ozone en cas d'orage. En revanche, la sensibilité des êtres humains aux accélérations lors d'évolutions (montée/descente, virage) n'est pas fiable, ce qui peut engendrer des problèmes d'orientation spatiale.

Systèmes d'aide au pilotage

Les aides au pilotage concernant l'accident sont :

- les calculateurs de commande de vol, qui interprète les ordres du pilote au sidestick pour orienter les gouvernes. L'objectif du système est de rendre le vol sûr, économique et confortable pour les passagers [1] ;
- le pilote et la poussée automatiques (resp. PA et A/THR), qui ont entre autres pour rôle de décharger les pilotes humains de la tâche d'atteindre et maintenir les consignes renseignées par les pilotes (consignes simples comme le cap ou plus élaborées comme une trajectoire d'approche)
- le directeur de vol (FD), qui donne des indications aux pilotes sur les actions à effectuer sur le PFD (cabrer, piquer, à g., à d.) avec les barres de tendance pour suivre les consignes de trajectoire retenues.

Les systèmes d'aide au pilotage ont plusieurs modes de fonctionnement. Ceux du calculateur de commande de vol sont appelés lois de pilotage. Ces lois définissent la présence ou non d'asservissements automatiques, la fonction de transfert entre les périphériques d'entrée et les systèmes physiques de l'avion, ainsi que l'existence ou non de protections contre des commandes qui feraient sortir l'avion du domaine de vol. Les changements d'une loi à l'autre se font à l'initiative des pilotes mais peuvent aussi être déclenchés automatiquement par des événements extérieurs (par ex. des pertes de mesure de vitesse). Dans le cas du vol AF447, les lois qui nous intéressent sont les lois « normale », « alternate » et « alternate 2 ».

Par exemple, en loi « normale », les traductions des demandes en inclinaison par le sidestick se font en fonction d'objectifs de sécurité mais aussi de confort. En loi « alternate 2 », l'angle latéral du sidestick contrôle directement les ailerons. Les inclinaisons du sidestick n'ont donc pas les mêmes effets sur le comportement de l'avion. Les automatismes offrent aussi des protections : il est difficile pour les pilotes de faire sortir l'avion du domaine de vol. Ainsi, en loi normale il faut insister sur le sidestick pour dépasser 33° d'inclinaison, alors que ces protections sont levées en loi alternate 2. Les protections sont d'autant plus utiles qu'en haute altitude le domaine de vol est réduit : la vitesse de croisière est relativement proche tant du décrochage bas que du décrochage haut. Les marges entre l'incidence de décrochage d'un côté et de la survitesse de l'autre côté sont limitées. Lorsque les protections sont désactivées, il appartient alors aux pilotes seuls de respecter le domaine de vol. Ils disposent pour cela d'instruments de bord qui affichent les grandeurs pertinentes (vitesse, attitude de l'avion, etc), d'alarmes sonores et visuelles, d'une documentation fournissant des tableaux de valeurs pré-calculées, et de valeurs à connaître de mémoire (memory items) dans le cadre de procédures d'urgence, comme par exemple la procédure « IAS (Indicated Aircraft Speed) » douteuse qui aurait dû être appliquée dans le cas présent.

LE RAPPORT DU BEA ET L'IHM

Dans cet article, nous sélectionnons uniquement les faits et analyses du BEA qui nous semblent accessibles à la modélisation en utilisant le corpus théorique de l'interaction

homme-machine. Ce choix impose des précautions de lecture. Tout d'abord, malgré nos efforts notre sélection peut présenter des erreurs ou mauvaises interprétations. Le lecteur est invité à se référer au rapport du BEA en cas de doute. Ensuite, notre sélection peut induire le lecteur en erreur sur les causes de l'accident. Le lecteur est invité à garder en tête qu'il existe d'autres faits importants, absents seulement parce qu'ils ne nous semblent pas accessibles aux modèles de l'IHM.

En particulier, nous nous limitons à une phase précise du vol, celle qui débute à 2 h 10 min 05 par le givrage des sondes Pitot et le débrayage du sous-système de Pilote Automatique. Certaines causes de l'accident trouvent leurs racines bien avant cet instant, et sont liées au fonctionnement du système homme-machine, qu'il s'agisse de l'entraînement des pilotes ou des choix d'organisation de l'équipage pour ce vol. Mais les dernières minutes sont celles qui relèvent le plus clairement de l'interaction homme-machine et de ses modèles.

Nous reproduisons ci-dessous le synopsis de l'accident et la synthèse des faits et de l'analyse des causes par le BEA. Puis, dans les sections suivantes, nous nous focaliserons sur une série de phénomènes mis en évidence dans l'analyse et nous les relierons à des théories de l'IHM.

Synopsis de l'accident

Vers 2 h 08, l'équipage effectue une déviation de 12 degrés vers la gauche, probablement pour éviter des échos détectés par le radar météo. A 2 h 10 min 05, vraisemblablement à la suite de l'obstruction des sondes Pitot par des cristaux de glace, les indications de vitesse deviennent erronées et des automatismes se désengagent. La trajectoire de l'avion n'est pas maîtrisée par les deux copilotes. Ils sont rejoints 1 minute 30 plus tard par le commandant de bord, alors que l'avion est dans une situation de décrochage qui se prolonge jusqu'à la collision avec la mer, à 2 h 14 min 28. (p19)

Analyse par le BEA

L'objectif des travaux sur les facteurs humains a été d'examiner le sous-ensemble de ces dispositions qui concernent les comportements et compétences attendus des équipages pour la situation rencontrée. [...] Au-delà de la simple découverte d'une explication psychologiquement probable, vraisemblable ou plausible aux comportements constatés, il s'agissait aussi d'évaluer le degré de spécificité ou de généricité des réponses comportementales constatées : sont-elles spécifiques à cet équipage particulier, partagées par tous les équipages de la compagnie ou encore généralisables à tous les équipages ? (p107)

Faits établis par l'enquête :

- Les systèmes de l'avion ont détecté une incohérence des vitesses mesurées. La loi de commande de vol s'est reconfigurée en loi alternate 2B.
- Aucun message de panne n'indique clairement à l'ECAM la détection par les systèmes d'une incohérence des vitesses mesurées.

- Les pilotes ont détecté une anomalie par l'alarme de déconnexion du pilote automatique qui les a surpris.
- Bien qu'ayant identifié et annoncé la perte des indications de vitesse, aucun des deux copilotes n'a appelé la procédure « IAS douteuse ».
- Les directeurs de vol n'ont pas été désengagés. La vitesse affichée sur le PFD gauche a été erronée pendant 29 secondes, la vitesse de l'ISIS pendant 54 secondes, et la vitesse affichée sur le PFD droit pendant au maximum 61 secondes. En moins d'une minute après le désengagement du pilote automatique, l'avion est sorti de son domaine de vol à la suite d'actions de pilotage inadaptes.
- Les barres de tendance ont disparu puis réapparu à plusieurs reprises en changeant plusieurs fois de mode.
- L'approche du décrochage a été caractérisée par l'activation de l'alarme puis l'apparition de buffet.
- En l'absence de la présentation des vitesses limites sur le bandeau de vitesse du PFD, l'alarme de décrochage sonore n'est confirmée par aucune indication visuelle spécifique.
- L'alarme de décrochage a retenti de façon continue pendant 54 secondes.
- Aucun des pilotes n'a fait référence à l'alarme de décrochage ni au buffet.
- L'incidence est le paramètre qui permet d'activer l'alarme de décrochage ; si les valeurs d'incidences deviennent invalides, l'alarme s'arrête.
- Par conception, lorsque les valeurs de vitesse sont mesurées inférieures à 60 kt, les valeurs d'incidence mesurées sont invalidées.
- L'incidence de l'avion n'est pas directement présentée aux pilotes. (p203)

Causes de l'accident (extraits) :

- L'équipage, progressivement déstructuré, n'a vraisemblablement jamais compris qu'il était confronté à une « simple » perte des trois sources anémométriques.
- Dans son état actuel, la reconnaissance de l'alarme de décrochage, même associée au buffet, suppose que l'équipage attribue à l'alarme une « légitimité » minimum. [...] Lorsqu'une action de l'équipage est attendue, il est toujours supposé qu'il aura une capacité de maîtrise initiale de la trajectoire et de diagnostic rapide permettant d'identifier la bonne entrée dans le dictionnaire de procédures. Un équipage peut être confronté à une situation imprévue entraînant une perte momentanée mais profonde de compréhension. [...] Lors de cet événement, l'incapacité à maîtriser initialement la trajectoire a aussi rendu impossible la compréhension de la situation et l'accès à la solution prévue. (p205)

L'accident résulte de la succession des événements suivants :

- l'incohérence temporaire entre les vitesses mesurées, vraisemblablement à la suite de l'obstruction des sondes Pitot par des cristaux de glace ayant entraîné notamment la déconnexion du pilote automatique et le passage en loi alternante ;
- les actions inappropriées sur les commandes déstabilisant la trajectoire ;
- l'absence de lien, de la part de l'équipage, entre la perte des vitesses annoncée et la procédure adaptée ;
- l'identification tardive par le pilote non-volant (PNF) de l'écart de trajectoire et la correction insuffisante par le pilote volant (PF) ;
- la non identification par l'équipage de l'approche du décrochage, l'absence de réaction immédiate et la sortie du domaine de vol ;
- l'absence de diagnostic de la part de l'équipage de la situation de décrochage et en conséquence l'absence d'actions permettant de la récupérer. (p205)

Ces événements peuvent trouver leurs explications dans la combinaison des facteurs suivants :

- les mécanismes de retour d'expérience de l'ensemble des acteurs qui n'ont pas permis [...] de détecter la non-application récurrente de la procédure relative aux pertes d'informations de vitesses et d'y remédier,
- un travail en équipage affaibli par :
 - l'incompréhension de la situation à la déconnexion du PA,
 - l'absence d'indication claire dans le poste de pilotage de l'incohérence des vitesses identifiée par les calculateurs ; (p207)

De cette analyse faite par le BEA nous avons sélectionné cinq phénomènes (noté de P1 à P5) qui nous semblent à la fois jouer un rôle significatif dans l'analyse et constituer des exemples d'application de modèles disponibles.

P1 : NON DÉTECTION DES CHANGEMENTS DE MODE

En cas d'incident, il est attendu du PF et du PNF de reprendre en main la stabilité de l'avion, puis d'analyser l'événement. Ici, il existe un doute sur l'identification du changement de loi.

La saillance de l'anomalie de vitesse étant très faible devant celle de la déconnexion du pilote automatique, l'équipage détecte un problème par cette déconnexion et non par la perte des indications de vitesse. Pour les mêmes raisons de saillance, il est probable que l'équipage n'a pas encore perçu le passage en loi alternante et la déconnexion de l'A/THR. (p180)

L'équipage va néanmoins se construire un début de représentation mentale de la situation une dizaine de secondes

après la déconnexion du PA par l'identification d'une anomalie des informations de vitesse. Toutefois il ne précise pas le nombre de sources de vitesses perdues. L'annonce de la perte des vitesses est quasi simultanée par les deux pilotes. (p181)

Lorsque l'une des trois vitesses s'écarte trop des deux autres, elle est automatiquement rejetée par les PRIM et la valeur votée devient alors la moyenne des deux valeurs restantes. Mais si l'écart entre ces deux valeurs restantes devient trop grand, les PRIM les rejettent et la loi de pilotage devient alternante 2. (p40)

Il n'existe cependant aucune indication explicite, en dehors du drapeau SPD LIM rouge à côté du bandeau de vitesse, du niveau de loi alternante dans laquelle se trouve l'avion. Le message ECAM associé au passage en loi alternante, quelle qu'elle soit, porte la mention « PROT LOST » ; cependant toutes les protections ne sont pas perdues, puisque la protection en facteur de charge reste disponible, et que des protections réduites peuvent également exister. L'identification précise des conséquences d'une reconfiguration en loi alternante est donc compliquée. (p193)

Ces problèmes réfèrent à la gestion des modes d'une interface. Un mode est un état de l'interface dans lequel les mêmes actions des utilisateurs sont interprétées différemment des autres modes. Les modes posent deux problèmes aux utilisateurs : celui de leur perception, et celui de la mémorisation des actions possibles et de leurs effets. La perception du mode courant est d'autant plus problématique si l'initiative du changement de mode est celle de l'appareil et non des opérateurs humains, et que ce changement n'est pas perçu. Pour l'AF447, les pilotes n'ont pas perçu immédiatement le passage en loi (ou mode) alternante 2. Ils n'ont pas non plus inféré un tel déclenchement, qui dépend notamment du nombre de sources de vitesse perdues : 1 vitesse perdue → alternante, 2 vitesses perdues → alternante 2.

Les concepteurs d'IHM préconisent d'éviter les modes [16, 19] dans la mesure du possible car ils sont source de nombreuses erreurs. Pourtant, le vol d'un avion présente une telle combinatoire qu'il est difficile d'éviter l'utilisation de modes. Il serait donc utile d'approfondir le rôle des modes pour les systèmes complexes : est-il possible de les éliminer, et dans la négative comment rendre leur perception plus immédiate ?

P2 : ADAPTATION AU CHANGEMENT DE LOI

Dès les premières secondes après le désengagement du pilote automatique, le PF a dû reprendre le pilotage de l'attitude de l'avion et s'adapter au changement de loi de pilotage, sans avoir pris conscience de ce changement.

Le PF est immédiatement absorbé par le pilotage en roulis dont les oscillations peuvent s'expliquer par : une action initiale sur le mini-manche, de forte amplitude sous l'effet de la surprise ; l'entretien des oscillations, le temps d'adapter son pilotage à haute altitude, qui plus est dans une loi de pilotage inhabituelle en roulis (loi directe).

Le caractère excessif des actions du PF peut s'expliquer par la surprise et la charge émotionnelle à la déconnexion du PA, amplifiées par l'absence de formation pratique des équipages au vol à haute altitude, de surcroît avec des lois de commande de vol inhabituelles. (p179)

Dans le cas de l'accident, le PF a cherché à contrôler le roulis, même si l'amplitude de ses actions a finalement entreteint ces mouvements ; l'action à cabrer relativement forte qu'il applique dans le même temps a pu, entre autres hypothèses, trouver son origine dans une certaine difficulté à intégrer la différence de type de loi de commande et donc la différence de type de pilotage à adopter entre les deux axes. (p194)

Les changements de lois de pilotage correspondent à des changements de fonctions de transfert. Les fonctions de transfert réfèrent aux relations entre les mouvements dans l'espace de contrôle (ici le sidestick) et l'espace de résultat (ici l'attitude de l'avion)[3]. Nous n'avons pas connaissance de travaux sur les performances de l'adaptation « dynamique » des utilisateurs à un changement de fonction de transfert, qu'ils en soient conscients ou pas. Mieux comprendre ces performances et les coûts cognitifs induits pourrait aider à préparer les pilotes à ces changements, ou à revoir des choix de conception si nécessaire.

P3 : DIFFICULTÉS DE PERCEPTION

Le BEA émet l'hypothèse que durant les premières secondes après la perte des informations de vitesse, les pilotes ont pu éprouver des difficultés à percevoir et organiser les informations émises en leur direction.

Saturation audio et tentative pour y remédier

Il est à noter que, pendant cette période de 46 secondes entre la déconnexion du pilote automatique et l'activation de l'alarme STALL 2 [2 pour deuxième activation], l'alarme C-chord [qui indique une différence d'altitude mesurée et prévue] a sonné sur une durée cumulée de trente-quatre secondes, dont trente-et-une de manière continue, et l'alarme STALL 1 pendant deux secondes. L'alarme C-chord a donc saturé l'environnement sonore du poste de pilotage. Elle n'a pas été annulée par l'équipage. Cette ambiance sonore a certainement contribué à altérer la réponse de l'équipage à la situation. A 2 h 10 min 47, l'un des membres d'équipage a annulé l'alarme Master Caution qui était active depuis 2 h 10 min 05. Il a pour cela appuyé sur le bouton poussoir au FCU, qui était alors allumé en ambre. [L']alarme [C-Chord] est annulable par appui sur le bouton poussoir Master Warning, qui n'est pas allumé, situé juste à côté du bouton poussoir Master Caution. Il est possible que cette action d'appui sur le bouton Master Caution ait été une action réflexe à la constatation visuelle que le bouton était allumé. Il est cependant aussi envisageable qu'elle ait été faite dans l'intention d'annuler l'alarme C-Chord afin de désaturer l'ambiance sonore dans le poste de pilotage. (p194)

Signifiant

Le problème de l'appui potentiellement erroné réfère au concept de « signifiant » [15]. Les signifiants (*signifiers*) réfèrent à tout indicateur perceptible (marque visuelle,

son, toucher, goût, force) qui communique un comportement adéquat à une personne. Les signifiants peuvent être des signes, des labels, et des dessins placés à des endroits appropriés comme des textes « pousser », « tirer » ou « sortie » sur les portes, ou des flèches et des diagrammes qui indiquent ce qui peut être manipulés et la direction de manipulation, ou d'autres instructions. Un bon signifiant communique aux gens qui utilisent un appareil son objectif, sa structure et son mode opératoire.

Ici, l'alarme C-Chord est une alarme de niveau « Warning », mais le bouton « Master Warning » qui permet de les couper n'est pas allumé. Le bouton « Master Caution » est lui allumé, et a pu signifier à l'utilisateur qu'il permettait d'éteindre l'alarme. De plus, le résultat infructueux de son action, contraire au fonctionnement attendu, a peut-être perturbé l'utilisateur.

Notification

Les alarmes sonores sont quant à elles des « notifications ». Une notification est un signifiant temporaire (passant d'un état non-perceptible à perceptible) déclenché par un sous-système (généralement automatique) à destination d'un autre sous-système (généralement humain). Les notifications sont nécessaires dans le cas où les actions déclenchées par l'utilisateur ont des effets dans un futur au-delà de leur mémoire immédiate, ou dans les interfaces de type supervision, pour lesquelles le monde physique évolue de façon indépendante des actions de l'utilisateur. Sans les notifications, l'utilisateur est engagé dans un processus de surveillance, potentiellement coûteux en ressources.

Une notification doit capter l'attention de l'utilisateur [20]. Elles doivent donc utiliser un signifiant compatible avec les capacités de perception, notamment en contexte dégradé : ajout d'une marque visuelle vibrante en périphérie d'une taille optique élevée, son couvrant les sons d'environnement (moteur, vent) et vibrant en hauteur, vibration d'un manche de sidestick, etc [5]. Une fois l'attention détectée, la notification ne doit pas gêner l'activité de l'utilisateur. Si plusieurs notifications se déroulent simultanément, elles doivent être hiérarchisées par l'appareil. Dans le cas du cockpit de l'A330, cette fonction est assurée par le choix de l'alarme sonore à diffuser (alternance entre C-Chord et STALL), et par l'affichage de messages dans l'ECAM avec des codes couleur correspondant au niveau d'alarme. Enfin, une notification se doit d'être informative et doit convoyer un message détaillé à l'utilisateur.

Signifiants, Rétroaction et Transaction

les conditions de vol de nuit en IMC rendent plus difficile la surveillance des attitudes (assiette en particulier). (p180)

Les signifiants peuvent être délibérés et intentionnels (un message « pousser » sur une porte) ou accidentel et non intentionnel. Dans le cas des cockpits, les pilotes utilisent l'horizon artificiel pour percevoir l'attitude de l'avion mais peuvent s'aider de la vue extérieure. Pour l'AF447, la nuit et l'absence de lumières au-dessus de l'océan ne permettent pas de profiter de ce signifiant.

l'approche d'un décrochage sur un avion classique est toujours associée à un effort à cabrer de plus en plus prononcé. Ce n'est pas le cas sur l'A330 en loi alternante. Il en résulte notamment que dans cette loi de pilotage l'avion, mis dans une configuration où la poussée n'est pas suffisante pour maintenir la vitesse sur la trajectoire, finirait par décrocher sans action sur le manche. Il apparaît que cette absence de stabilité statique positive a pu contribuer à l'absence d'identification par le PF de l'approche du décrochage. (p194)

le PNF dit « commandes à gauche », prend la priorité et donne une action brève en butée à gauche ; le PF reprend presque immédiatement la priorité sans aucune annonce et continue à piloter. (p25)

Il est à noter que les actions appliquées par un pilote sur un mini-manche ne sont pas facilement observables par l'autre. (p180)

La rétroaction (*feedback*) est le signifiant qui communique le résultat d'une action : allumage d'un voyant, émission d'un « clic » etc.¹ Les commandes électriques de l'A330 évitent aux pilotes des efforts physiques pour piloter l'appareil. Il n'y a donc pas de rétroaction haptique sur le sidestick liée aux phénomènes physiques (« stabilité statique positive »), à moins que la rétroaction soit reproduite par des systèmes à retour d'efforts. Ce n'est pas le cas des A330, et ceci a pu constituer une des raisons identifiées par le BEA à la non-identification de l'approche du décrochage. Par ailleurs, il n'y a pas de rétroaction dans le sidestick du PNF des actions du PF sur son sidestick [15]. Ceci réfère à la notion de collecticiels, ces systèmes dont les interactions sont conçues pour favoriser la conscience des activités des co-équipiers, et en particulier la transaction/*feedthrough* [7] (dont la terminologie est construite sur le modèle du terme rétroaction/*feedback* : « renseigner à travers l'interface » envers un équipier plutôt que « renseigner en retour » envers soi-même).

Evolutions de l'ECAM

De la déconnexion du pilote automatique à l'activation de l'alarme STALL 2, de nombreux messages se sont affichés à l'ECAM. Aucun de ces messages n'a aidé l'équipage à identifier le problème lié à l'anomalie de vitesse. Par ailleurs, la gestion des priorités des différents messages a provoqué des changements d'affichage rapides et ainsi rendu encore plus compliquée l'analyse et la compréhension de la situation. (p194)

L'ECAM est l'appareil qui affiche les notifications en cours. L'arrivée de nouvelles notifications modifie l'affichage, et notamment l'ordre des messages selon la dimension Y de l'écran. Les changements d'affichage se font immédiatement, ce qui rend difficile la perception de l'occurrence d'un changement si celui-ci a lieu alors que les pilotes ne regardent pas l'ECAM (vision périphérique) [17], et ce qui rend difficile la compréhension du réordonnement des messages même si les utilisateurs sont en train de le regarder [14, 17].

¹ Norman semble assimiler les notifications et les rétroactions [15]. Nous divergeons sur ce point et les distinguons : la rétroaction doit être immédiate pour être comprise comme telle.

P4 : PILOTAGE DE L'AVION

Au-delà de la reprise en main des premières secondes, le PF doit reprendre à sa charge le pilotage de l'avion dans le but de retrouver une situation stable et une trajectoire sûre, laissant le temps de choisir une procédure d'urgence ou d'explorer le problème. Ici, le PF n'est pas parvenu à atteindre une situation stable.

En premier lieu, on attend de l'équipage qu'il assure le contrôle de l'avion et le suivi de la trajectoire. (p108)

Les premières perturbations des vitesses 1 et 2 interviennent vers 2 h 10 min 04, engendrant la déconnexion du pilote automatique, marquée par une alarme visuelle et sonore (charge de cavalerie). L'équipage ne perçoit pas forcément ces pertes furtives des informations de vitesses et les baisses d'altitudes associées. La première diminution prolongée (au moins 5 secondes) de la vitesse sur le PFD droit débute au plus tard à 2 h 10 min 07. Elle entraîne une baisse de l'altitude affichée sur ce PFD d'environ 330 ft. A partir de 2 h 10 min 08, la vitesse devient aberrante à gauche. La saillance de l'anomalie de vitesse étant très faible devant celle de la déconnexion du pilote automatique, l'équipage détecte un problème par cette déconnexion et non par la perte des indications de vitesse. Par une action réflexe, il réagit comme il l'a appris en reprenant en pilotage manuel (annonce « j'ai les commandes » par le PF, accusé réception par le PNF).

(p180)

Si la première réaction excessive à cabrer du PF peut donc être comprise, il n'en va pas de même de la persistance de cette action qui va provoquer un important écart de trajectoire verticale. [...] Il reste un certain nombre d'éléments d'explication possible : une focalisation de l'attention sur le roulis, la vitesse, ou encore sur l'ECAM, au détriment de la surveillance de l'assiette ; un déclenchement plus ou moins conscient, sous l'effet de la surprise et du stress, du plan d'action (monter) souhaité par le PF avant la déconnexion du pilote automatique ; [...]

(p179)

Les oscillations d'assiette dans les secondes qui suivent l'activation de l'alarme de décrochage montrent que le pilotage de l'avion était manifestement délicat et a probablement monopolisé l'attention du PF. Dans cette phase, la maquette avion [représentation de référence de l'avion sur l'horizon artificiel] sur le PFD était proche de la barre horizontale du directeur de vol [FD], en moyenne légèrement supérieure. Le PF a probablement cherché à suivre l'évolution de cette barre, sans avoir intégré le changement du mode longitudinal engagé.

(p187)

Note : la procédure « vol avec IAS douteuse » prévoit de désactiver les FD, pour éviter la présentation d'ordres potentiellement non pertinents.

(p188)

Afin de reprendre le contrôle de l'appareil, le pilote exécute des actions et évalue le résultat de ces actions en percevant le monde qui l'entoure. Ces processus peuvent être modélisés avec le « cycle de l'action » [15] (fig. 4). Quand des personnes utilisent un appareil, elles doivent

faire face à deux « gouffres » (*gulfs*) : le Gouffre de l'Exécution, où elles essaient de comprendre comment opérer, et le gouffre de l'Evaluation, où elles essaient de comprendre ce qui est arrivé après avoir opéré.

La figure 4 montre deux instanciations de ce modèle pour le PF. Le premier cycle concerne la déconnexion du pilote automatique : le PF la perçoit avec l'alarme sonore C-Chord, mais il ne perçoit pas la perte des vitesses. Le deuxième cycle montre l'activité du pilote pour maîtriser la trajectoire. L'alarme STALL 1 retentit tronquée puis s'arrête, pendant que le PF suit les indications du directeur de vol affichées sur le PFD : tout en contrant les inclinaisons, il met à cabrer l'appareil.

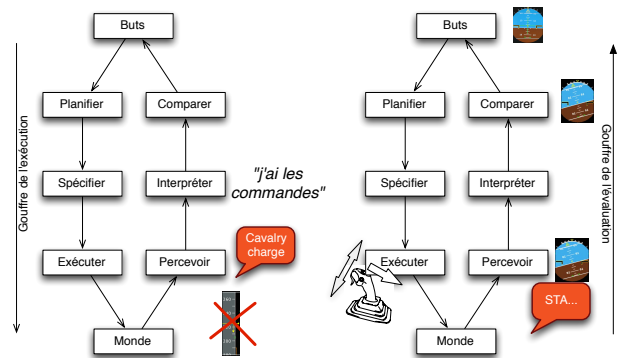


Figure 4. Cycles de l'action du PF

P5 : RECHERCHE DU PROBLÈME (OU ABDUCTION)

Le cycle de l'action est une simplification de la réalité mais constitue cependant un cadre utile pour comprendre les actions humaines. En particulier, en cas de fonctionnement inattendu, les utilisateurs remettent en cause les éléments afférents aux concepts cités par Norman : signifiants, rétroactions, notifications. C'est notamment durant ces situations qu'ils cherchent à comprendre les causes des notifications présentes. Ainsi, il est attendu des pilotes qu'ils explorent la nature du problème pour identifier les actions permettant d'y remédier, voire d'éventuelles procédures d'urgences à appliquer.

Il est alors prévu que l'équipage détectera l'anomalie, qu'il « fera sens » éventuellement de cette détection [...] A partir des informations disponibles à l'ECAM, l'équipage doit analyser et confirmer la nature de la défaillance avant d'entreprendre toute action de traitement de la panne [...] L'application de règles suppose non seulement leur connaissance, mais aussi la reconnaissance de leurs conditions d'applicabilité, et donc une bonne identification plus une certaine interprétation de l'anomalie. La construction d'une réponse par appel aux connaissances suppose une incorporation de l'anomalie dans la représentation mentale de la situation, qui peut passer par une destruction/reconstruction de celle-ci, très coûteuse en ressources et chronophage. [...] Ainsi, la perception correcte de la situation par un équipage, qui permet d'améliorer la fiabilité et la rapidité de diagnostic et de décision, est liée non seulement à la manière dont la situation est présentée à cet équipage (interfaces, paramètres) mais aussi à sa formation et à son expérience.

(p108)

Un processus d'abduction

L'activité que doit conduire le PNF est un processus d'abduction : « le processus permettant d'expliquer un phénomène ou une observation à partir de certains faits, événements ou lois. [...] Dans le domaine médical par exemple, le diagnostic final explique les signes et les symptômes d'un patient en supposant un dysfonctionnement tel qu'une maladie ou une fracture » [11]. Le BEA exprime les éléments nécessaires à l'aboutissement d'un tel processus :

[il faut] que [l]es signes [du problème] soient crédibles et pertinents ;
[...] que les indications disponibles relatives à l'anomalie soient très rapidement identifiables afin que les éventuelles actions immédiates de mémoire visant à stabiliser la situation soient déclenchées, ou que l'identification de la procédure applicable soit faite correctement. En particulier, il est important que les interfaces habituellement porteuses des informations d'anomalie affichent ou du moins permettent ce diagnostic initial, compte tenu des compétences minimales attendues d'un équipage.
[...] qu'il n'y ait pas de signaux ou d'informations disponibles suggérant des actions différentes ou incitant l'équipage à une reconstruction préalable de la compréhension de la situation. (p198)

Les pilotes cherchent donc à éliminer des signifiants douteux, ou à chercher des cohérences entre les signifiants, à comprendre les conséquences de leurs actions, voire à agir pour tester des hypothèses.

Crédibilité des signifiants

L'alarme STALL se déclenche une première fois (STALL 1) dès le débrayage du PA pendant 2 secondes.

De telles activations [de l'alarme de décrochage STALL 1] sont la conséquence de l'augmentation locale de l'incidence ; elles sont donc transitoires et se manifestent généralement par une alarme tronquée (seule la voix synthétique « STALL, STALL » résonne, parfois incomplète). Les événements antérieurs étudiés (alarme de décrochage dans le cadre d'une anomalie de vitesse en croisière) montrent cependant que d'autres équipages n'ont pas réagi comme attendu à la proximité du décrochage et ont eu tendance à considérer l'alarme comme non pertinente. De ce fait, il faut considérer que le comportement de l'équipage AF 447 est probablement reproductible en ce qui concerne la non-réaction à l'alarme STALL 1. De telles activations furtives peuvent paraître inappropriées et nuire à la crédibilité globale d'une alarme qui n'est quasiment jamais rencontrée par les équipages en qualification de type, en vol ou en entraînement. (p195)

Les activations furtives de l'alarme après la déconnexion du pilote automatique ont pu amener l'équipage à douter de sa crédibilité. De plus, la présence du directeur de vol conduisant à afficher une assiette à cabrer a pu conforter le PF dans l'idée que l'alarme de décrochage n'était pas pertinente. Lors des événements antérieurs étudiés, les équipages ont fréquemment fait part de leurs doutes sur la pertinence de l'alarme. (p187)

Invisibilité des signifiants

L'équipage est uniquement informé des conséquences du déclenchement de ces surveillances : déconnexion du PA et de l'A/THR, passage en loi alternante, etc. Aucun message de panne ne conduit à identifier l'origine de ces autres pannes, le rejet des ADR et des mesures de vitesse en particulier. (p194)

Incompréhension des conséquences

Cependant, le PF a pu assimiler l'activation de l'alarme [STALL] à une conséquence de la réduction de poussée qu'il avait appliquée quatre secondes plus tôt ; il aurait alors appliqué la pleine poussée pour revenir à la situation antérieure. Quelques secondes plus tard, le PF a verbalisé « je suis en TOGA, hein ? ». Soit il a douté d'avoir placé les commandes de poussée dans le cran TOGA [poussée assurant une montée sûre à basse altitude] comme il en avait l'intention, soit il n'a pas compris pourquoi cette action était inefficace pour arrêter l'alarme de décrochage. Ce second cas indiquerait alors que le PF a eu une représentation erronée du modèle de vol de l'avion, espérant une efficacité de la poussée TOGA à haute altitude et une assiette de douze degrés, similaire à celle délivrée dans les basses couches. Le résultat infructueux de ses actions a peut-être renforcé sa méfiance vis-à-vis de l'alarme. (p187)

Actions pour tester des hypothèses

Les deux pilotes exécutent des actions afin de confirmer ou d'infirmer des hypothèses.

La désactivation de la fonction THRUST LOCK par le PF est le signe d'une recherche d'information. Le PF a donc pu être saturé par son investissement immédiat et naturel dans la tentative de comprendre la situation, venu s'ajouter à sa charge déjà élevée de pilotage. (p182)

Le PNF a quant à lui actionné le système d'antigivrage de la voilure après avoir lu l'ECAM, ce qui laisse penser qu'il a alors perçu une problématique de givrage fort. Le bruit des cristaux de glace sur le pare-brise, assimilé à de la pluie par d'autres équipages, a pu le conforter dans la perception d'un risque associé. Les symptômes perçus ont pu apparaître à l'équipage comme des anomalies complémentaires à celle des indications de vitesse et sembler composer un syndrome global beaucoup plus complexe que la seule perte des indications anémométriques. (p183)

Il a également agi sur le rotacteur ATT/HDG et verbalisé cette action (« je te mets en ATT... »). Ce changement de source inertielle, qui a posteriori n'était pas nécessaire, peut indiquer que son diagnostic sur la panne n'était pas complètement délimité. Pour lui, les informations de vitesse étaient incohérentes mais il n'excluait peut-être pas que les informations inertielles le soient aussi. Après avoir modifié la source ADR, le « qu'est-ce que c'est que ça » du PNF semble marquer son incompréhension totale face au résultat de cette action car la vitesse affichée côté droit était toujours erronée. Il semble alors dépassé. (p184)

Ce processus d'abduction peut être modélisé par le cycle de l'action. Les pilotes ont effectué plusieurs cycles, durant lesquels ils ont agi et évalué les conséquences de leurs actions. Durant ces cycles, en émettant et testant des hypothèses de fonctionnement, ils reconstruisent ce qu'on appelle un modèle conceptuel.

Modèle conceptuel

Un modèle conceptuel est une explication, généralement très simplifiée, du fonctionnement d'un appareil [15]. Par exemple, les interfaces informatiques reposent sur un modèle conceptuel basé sur des fichiers et des répertoires. Les modèles conceptuels fournissent une compréhension de l'appareil, des prédictions sur comment l'appareil va se comporter, et des solutions quand ce qui se passe ne concorde pas avec ce qui a été planifié. Ils sont considérés comme l'élément le plus déterminant de l'utilisabilité des systèmes interactifs [12, 9]. Sans utiliser cette dénomination, le BEA évoque les modèles conceptuels :

En l'absence d'indication de vitesse fiable, la compréhension de la physique globale du vol à haute altitude par une approche synthétique des bilans énergétiques, équilibres de forces, plafonds de sustentation et propulsion, aurait pu considérablement aider les pilotes à anticiper la dégradation rapide de la situation et à prendre à temps la mesure corrective adéquate : la mise en descente. [...]

Le Manuel Aéronautique Complémentaire d'Air France expose d'une façon très détaillée, sur 38 pages, la physique du vol Haute Altitude avec des exemples de cas réels. Ces connaissances font également partie de l'enseignement théorique censé être dispensé à un stade avancé de la formation d'un futur pilote de ligne (ATPL théorique, performances en QT, etc.). La trajectoire en montée, initialement plus ou moins volontaire de l'équipage mais maintenue trop longtemps, est probablement un indice d'une assimilation insuffisante de ces notions théoriques. (p190)

Les modèles conceptuels décrits dans les manuels techniques peuvent être complexes. Ils doivent cependant être compatibles avec les capacités cognitives des utilisateurs à qui ils sont destinés ainsi qu'avec les tâches qui leur incombent dans un contexte donné.

Modèle conceptuel de l'alarme de décrochage

La surdité inattentionnelle [6] a été évoquée à propos de l'alarme de décrochage, mais le BEA n'a pas retenu cette explication. Plusieurs passages du rapport suggèrent plutôt qu'un des modèles conceptuels utilisés par les pilotes n'était pas suffisant pour comprendre la situation.

Jusqu'à la fin du vol, les valeurs d'incidence sont redevenues successivement valides et invalides. A chaque fois qu'au moins une valeur redevenait valide, l'alarme de décrochage se réactivait, et à chaque fois que les incidences étaient invalides, l'alarme s'arrêtait. [...]

Plusieurs actions à piquer ont provoqué une diminution de l'assiette et de l'incidence dont les valeurs sont redevenues valides, de telle sorte qu'une action franche à piquer s'est traduite par la réactivation de l'alarme de décrochage. Il semble que le PF ait, au moins à deux reprises, réagi par une action à cabrer dont les conséquences ont

été une augmentation de l'incidence, une diminution des vitesses mesurées et, par conséquent, l'arrêt de l'alarme de décrochage. Jusqu'à la fin du vol, aucune valeur d'incidence valide n'a été inférieure à 35°. (p196)

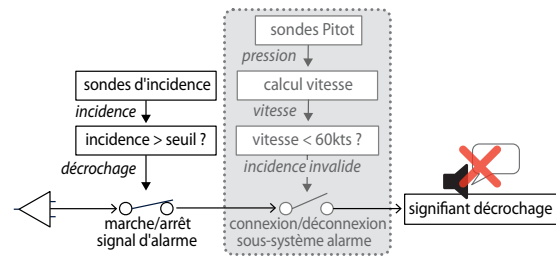


Figure 5. Modèles conceptuels du fonctionnement de l'alarme de décrochage (non-grisé : modèle du PF, grisé : modèle plus réel)

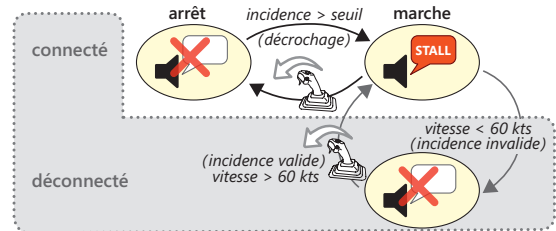


Figure 6. Modèles conceptuels des états et transitions du signifiant de décrochage (non-grisé : modèle du PF, grisé : modèle plus réel)

La fig. 5 représente deux modèles conceptuels du fonctionnement de l'alarme de décrochage : le modèle supposé du PF, et un modèle plus réaliste qui illustre l'absence de signifiant en cas de vitesse faible. La fig. 6 montre les modèles conceptuels des états et transitions du signifiant de décrochage. Le PF a peut-être pensé que l'alarme passe de l'état « arrêt » à « marche » quand elle retentit, alors qu'elle est passée de l'état « marche-déconnecté » à « marche-connecté ». Les deux états « arrêt-connecté » et « marche-déconnecté » ont des signifiants similaires (silence), ce qui les rend indiscernables. Le PF a peut-être constaté que les conséquences de ses actions sont inverses de ce que prévoit son modèle conceptuel : en général, les actions à piquer sont censées faire sortir du décrochage, alors qu'elles ont signalé ici une entrée de décrochage. Inférant peut-être que son action à piquer déclenche un décrochage, le PF stoppe son action en cabrant, ce qui coupe l'alarme en la faisant passer à l'état « marche-déconnecté » (et non à « arrêt »).

Mais au-delà de l'ergonomie de l'alarme, il est probable que la présentation d'une information synthétique sur la physique globale du vol (incidence, bilan énergétique, domaine de vol) pourrait aider les pilotes à « faire sens » de l'alarme et à prendre à temps les mesures correctives adéquates. [...]

En résumé les facteurs suivants sont de nature à diminuer la performance attendue d'un équipage : l'absence de description du fonctionnement de l'alarme de décrochage (schéma de principe ou ordre de grandeur des seuils, par exemple) dans la documentation ; (p196)

En conséquence, le BEA recommande que : l'AESA impose de revoir les conditions de fonctionnement de l'alarme de décrochage en vol lorsque les mesures de vitesse sont très faibles. (p218)

Ces constats suggèrent que le métier de pilote nécessite, pour les situations dégradées, des compétences semblables à celles des médecins, des chercheurs et des ingénieurs : la capacité à sélectionner des faits, remettre en cause leur modèle mental, construire un nouveau modèle plausible, et en déduire un plan d'action et de validation. Ceci est à rapprocher du programme de formation des pilotes, qui comprend des notions relativement avancées de physique du vol et de fonctionnement des systèmes : à quoi bon posséder ces notions si ce n'est pour les utiliser dans l'analyse de problèmes sortant du cadre habituel ? Mais alors, comment peut-on s'assurer au mieux que les signifiants disponibles en conditions dégradées permettent aux pilotes de mener un raisonnement correct dans un temps contraint ? Faut-il en déduire des critères de sélection supplémentaires, ou un entraînement spécifique ? Faut-il mener des recherches pour leur fournir des représentations de systèmes permettant de mieux gérer la complexité, comme cela est fait par ailleurs pour les ingénieurs [8] et les chercheurs [10] ?

CONCLUSION

Dans cet article, nous avons sélectionné des éléments d'analyse liés à l'IHM dans le rapport du BEA sur l'accident du vol AF447, nous avons assimilé et organisé ces éléments en axes connus de l'IHM, puis traduits dans des cadres conceptuels de l'IHM et de l'activité scientifique.

L'adéquation entre les données et ces cadres conceptuels fait du rapport du BEA un support intéressant pour l'enseignement de l'IHM. Il peut en particulier illustrer pour de futurs concepteurs l'importance d'éliminer dans la conception d'un système homme-machine tous les facteurs d'interprétation erronée de la situation par les opérateurs humains.

Notre travail peut aussi ouvrir de nouvelles opportunités de recherche. Malgré les efforts en cours [4, 13], les connaissances en IHM sont encore insuffisantes pour isoler de manière formelle l'ensemble des défaillances d'un système homme-machine, notamment celles évoquées ici. A plus forte raison, il reste difficile pour un ingénieur d'identifier toutes les configurations défavorables. En revanche, les connaissances nous semblent ici suffisantes pour organiser utilement une partie des données disponibles, voire pour permettre de formuler des hypothèses ou des théories complémentaires. C'est donc un outil possible pour progresser vers un objectif important pour l'IHM en aéronautique : des cadres théoriques qui permettent aux ingénieurs de mieux anticiper, dans un environnement non maîtrisé, le comportement des systèmes homme-machine qu'ils conçoivent [18].

Les phénomènes que nous avons reformulés dans cet article correspondent à une large gamme de thèmes de recherche, depuis les fonctions de transfert dans la boucle perception-action jusqu'au support à l'analyse de systèmes complexes. L'analyse de l'accident de l'AF447 peut donc participer à l'évaluation des futurs modèles théoriques qui seront proposés sur chacun de ces sujets : permettent-ils de décrire plus complètement les phénomènes à l'œuvre dans l'accident ?

Réciproquement, la recherche en IHM peut contribuer à renforcer la sécurité, en utilisant le rapport comme source d'inspiration. L'accident du vol AF447 nous rappelle en particulier l'importance d'éliminer les facteurs d'interprétation erronée de la situation. Raisonner sur la tâche d'abduction à laquelle ont été confrontés les pilotes peut fournir un nouvel angle pour les assister dans leur rôle de dernier recours : comment une IHM peut-elle assister un pilote - comme on assiste un ingénieur ou un scientifique, la contrainte temporelle en plus - dans la construction d'un raisonnement ?

BIBLIOGRAPHIE

1. Airbus. *A330 Flight deck and systems briefing for pilots*, 1999.
2. Bureau d'Enquêtes et d'Analyse. Rapport final - Accident survenu le 1er juin 2009 à l'Airbus A330-203 immatriculé F-GZCP exploité par Air France vol AF 447 Rio de Janeiro - Paris. Tech. rep., République Française, Ministère de l'Ecologie, du Développement durable et de l'Énergie, 2012.
3. Casiez G. & Roussel N. No more bricolage!: Methods and tools to characterize, replicate and compare pointing transfer functions. In Proc. *UIST '11*, ACM (2011), 603–614.
4. Chandler F., Chang Y., Mosleh A., Marble J., Boring R. & Gertman D. Human reliability analysis methods: selection guidance for nasa. Tech. rep., NASA Office of Safety and Mission Assurance, Washington, DC, 2006.
5. Chatty S., Athènes S. & Bustico A. Votre attention s'il vous plaît ! Éléments d'un espace de conception de signaux visuels. In *Actes de IHM*, Cépaduès Editions (1999).
6. Dehais F., Causse M., Vachon F., Régis N., Menant E. & Tremblay S. Inattention deafness to auditory alarms in aviation: A safety issue. *Human Factors and Ergonomics Society*, in press.
7. Dix A. J. *CSCW - a framework*. 1994, 23–37.
8. Gaspard-Boulin H. & Conversy S. Exigences d'utilisabilité pour les outils d'ingénierie des exigences. In Proc. *IHM*, ACM (2014).
9. Henderson A. & Johnson J. *Conceptual Models: Core to Good Design*, 1st ed. Morgan & Claypool Publishers, 2011.
10. Keim D., Andrienko G., Fekete J.-D., Görg C., Kohlhammer J. & Melançon G. Visual analytics: Definition, process, and challenges. In *Information visualization*. Springer, 2008, 154–175.
11. Lecourt D., Ed. *Dictionnaire d'histoire et philosophie des sciences*. PUF, 1999.
12. Liddle D. Bringing design to software. ACM, New York, NY, USA, 1996, ch. Design of the Conceptual Model, 17–36.
13. Martinie C., Navarre D. & Palanque P. A multi-formalism approach for model-based dynamic distribution of UI of critical interactive systems. *Inter. J. of Hum.-Comp. Studies* 72, 1 (2014), 77 – 99.
14. Mertz C., Chatty S. & Vinot J.-L. The influence of design techniques on user interfaces: the digistraps experiment for air traffic control. In *Proceedings of HCI Aero* (2000).
15. Norman D. A. *The Design of Everyday Things*, revised and expanded edition ed. Basic Books, USA, 2013.
16. Raskin J. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley, 2000.
17. Schlienger C., Conversy S., Chatty S., Anquetil M. & Mertz C. Improving users comprehension of changes with animation and sound: an empirical assessment. In Proc. *of Interact 2007, LNCS*, vol. 4662, IFIP, Springer Verlag (2007), 207–220.
18. Speyer J.-J. Les facteurs humains contribueront-ils à l'aéronautique ? *La Jaune et la Rouge*, 535 (1998), 59–69.
19. Tesler L. A personal history of modeless text editing and cut/copy-paste. *interactions* 19, 4 (2012), 70–75.
20. Wickens C., Hollands J., Parasuraman R. & Banbury S. *Engineering Psychology & Human Performance*. Pearson Education, 2012.

Exigences d'utilisabilité pour les outils d'ingénierie des exigences

Hélène Gaspard-Boulinc
 Université de Toulouse - ENAC
 31055, Toulouse, France
 helene.gaspard-boulinc@enac.fr

Stéphane Conversy
 Université de Toulouse- ENAC
 31055, Toulouse, France
 stephane.conversy@enac.fr

RESUME

Les outils d'ingénierie des exigences sont nécessaires pour plusieurs raisons : ils permettent de gérer un nombre croissant d'informations, de maintenir la traçabilité entre exigences, solution et tests, d'évaluer l'impact d'un changement d'exigences sur la solution et les tests.

Cet article propose une analyse des pratiques en ingénierie système dans l'industrie aéronautique, centrée sur l'ingénierie des exigences. A partir de la réalisation d'enquêtes contextuelles, nous avons identifié des scénarios représentatifs de l'activité des ingénieurs système sur la rédaction et la gestion des exigences. L'analyse de ces scénarios nous a permis d'identifier des exigences en termes d'utilisabilité des outils, plus précises que « user-friendliness », qui alimenteront la conception de nouveaux outils d'ingénierie des exigences.

Mots Clés

Ingénierie système; exigence; utilisabilité.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. Voir <http://www.acm.org/about/class/1998/> pour la liste complète des catégories ACM.

INTRODUCTION

La complexité des systèmes conçus par les êtres humains a fait émerger le besoin d'un cadre améliorant la communication et la coopération parmi les parties qui créent et gèrent les composants d'un système. Par exemple, le projet ATLAS de missile balistique intercontinental dans les années 50 aux Etats-Unis [9] a impliqué 18 000 scientifiques, ingénieurs, experts techniques dans les universités et les industries. 70 000 personnes, de l'administration jusqu'à l'usine, réparties dans 22 entreprises, ont participé activement au projet. Ces chiffres montrent le nombre important d'acteurs mais également leur hétérogénéité, et donc la nécessité

d'un cadre de coordination entre ces acteurs.

La réponse est l'ingénierie système, définie désormais par des normes [11][12] : c'est une approche collaborative et interdisciplinaire dont l'objectif est de définir, concevoir, réaliser et maintenir des systèmes complexes, en cherchant à élaborer une solution optimisée entre performance, coût et délai.

La notion d'exigence – requirement en anglais- est un élément central de l'ingénierie système. Une exigence est « un énoncé qui traduit ou exprime un besoin et ses contraintes et conditions associées » [13]. Une exigence bien exprimée doit être vérifiable et techniquement réalisable, et doit avoir une seule interprétation possible. Ainsi, si l'exigence est exprimée en langage naturel, il est recommandé d'adopter des règles de syntaxe telles que [Condition] [Sujet] [Verbe] [Complément] [Contrainte], où le sujet est le système d'intérêt (**Figure 1**). Une exigence possède des attributs tels qu'un identifiant unique, une priorité, la partie prenante intéressée par l'exigence, la justification. Elle possède aussi des attributs de gestion (date de création, date de dernière modification, auteur, version du système implémentant l'exigence).

[sujet]	[verbe]	[complément]	[contrainte]
Le système atteint une vitesse de 180 km/h en 40 secondes			
[sujet]	[verbe]	[complément]	[contrainte]
Le système doit afficher les factures dues dans l'ordre ascendant de date de paiement.			
[sujet]	[verbe]	[complément]	[contrainte]
Le système doit rejouer les données radar et radio de façon synchronisée			

Figure 1: exemples d'exigences

Un référentiel (baseline) d'exigences est un ensemble d'exigences validées par les parties prenantes à une date donnée. Le référentiel guide les autres activités du processus d'ingénierie système et représente la définition du problème à résoudre. Il évolue dans le temps, du fait de l'évolutivité des exigences. Il doit lui aussi respecter des bonnes propriétés définies dans [13].

L'ingénierie des exigences (Requirement Engineering) désigne les activités d'ingénierie système qui visent à établir et maintenir les exigences à remplir par le système. Elle nécessite l'utilisation d'outils pour plusieurs raisons: les outils permettent d'organiser et

gérer un nombre important d'exigences, de maintenir la traçabilité entre exigences-solution-tests, d'évaluer l'impact d'un changement d'exigences sur la solution et les tests.

L'utilisabilité [16] d'un outil est primordiale dans l'efficacité de l'utilisateur : plus précisément, l'utilisabilité est « le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité (précision et degré d'achèvement des objectifs), efficacité (rapport entre les ressources dépensées et l'efficacité), et satisfaction (absence d'inconfort, attitudes positives dans l'utilisation du produit), dans un contexte d'utilisation spécifié ».

L'objectif de cet article est de préciser les exigences d'utilisabilité des outils d'ingénierie des exigences. Pour cela, nous décrivons d'abord les activités prescrites de l'ingénieur système. Nous proposons ensuite un état de l'art sur les travaux relatifs à la définition et l'évaluation de l'utilisabilité des outils d'ingénierie des exigences. Pour compléter ces travaux, nous adoptons une approche de conception participative, avec la réalisation d'enquêtes contextuelles dans l'industrie aéronautique. L'analyse de ces enquêtes nous a permis d'identifier des scénarios d'usage, avec les forces et les faiblesses des outils utilisés. Nous en déduisons des exigences d'utilisabilité sur les outils d'ingénierie des exigences, justifiées par nos scénarios.

ACTIVITES PRESCRITES DE L'INGENIEUR SYSTEME

La norme IEEE 1220 [12] propose un processus global d'ingénierie système divisé en 8 sous-processus centrés sur les activités techniques d'ingénierie (Figure 2) : analyse des exigences, validation des exigences, analyse fonctionnelle, vérification fonctionnelle, analyse système (à droite), synthèse, vérification du design, contrôle.

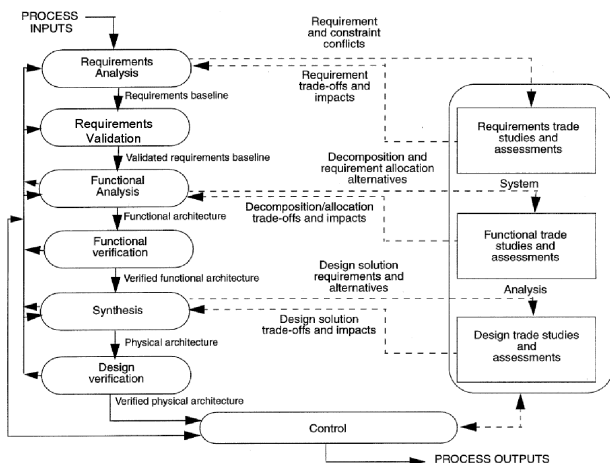
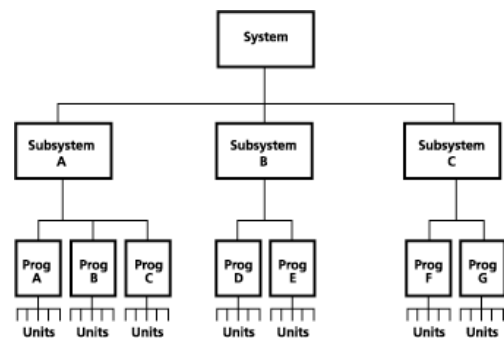


Figure 2: les processus d'ingénierie système selon la norme IEEE 1220-2005

Les processus sont reliés entre eux par des entrées-sorties. Les deux processus sur les exigences (Requirements Analysis et Requirements Validation)

produisent un référentiel vérifié d'exigences. Ce référentiel d'exigences représente la spécification du système à concevoir, en termes de fonctions, performances et interfaces avec l'environnement extérieur. Le référentiel d'exigences est en entrée des processus de conception fonctionnelle, qui produisent une architecture fonctionnelle vérifiée. Cette dernière est en entrée de l'analyse système qui va produire en sortie une architecture physique, c'est-à-dire une décomposition du système par composant, avec l'allocation des exigences par composant. Le processus de contrôle prend en entrée toutes les sorties des processus pour assurer leur documentation et sa mise à jour de façon ordonnée. Les flèches à gauche à double sens, ainsi que les flèches en pointillés, montrent la possibilité de pouvoir revenir sur le processus précédent de façon itérative tant qu'on identifie des problèmes (« variance and conflict » dans la norme).

Plus on avance dans la conception du système, plus les exigences sont affinées en exigences détaillées, pour atteindre une spécification précise des composants et des interactions entre composants auxquels les exigences sont allouées. Il en résulte une multiplication du nombre des exigences et une complexification des liens entre les différents éléments d'ingénierie (exigences, exigences détaillées, composants, interfaces entre composants, tests). Pour illustrer cela, nous pouvons reprendre l'exemple simple proposé par [24], d'un système composé de 3 sous-systèmes, avec 6 exigences (Figure 3a), où les exigences sont allouées par sous-systèmes (Figure 3b) et affiner en 21 exigences (Figure 3c).



SYSTEM REQUIREMENTS	SUBSYSTEM A	SUBSYSTEM B	SUBSYSTEM C
SYS 001	X	X	
SYS 002	X		X
SYS 003		X	
SYS 004	X	X	X
SYS 005			X
SYS 006	X	X	

SYSTEM REQUIREMENT	SYSTEM A REQUIREMENT	SYSTEM B REQUIREMENT	SYSTEM C REQUIREMENT
SYS 001	SSA 001 SSA 002	SSB 001	-
SYS 002	SSA 003 SSA 004 SSA 005	-	-
SYS 003	-	SSB 002 SSB 003	-
SYS 004	SSA 006 SSA 007	SSB 004 SSB 005 SSB 006	SSC 003
SYS 005	-	-	SSC 004 SSC 005
SYS 006	-	SSB 007 SSB 008	-

Figure 3a: décomposition d'un système en 3 sous-systèmes, figure 3b: allocation des exigences selon les sous-systèmes, figure 3c : exigences affinées par sous-système.

Mis à part les entrées/sorties qui lient les processus et l'aspect itératif entre processus, les normes ne précisent pas de cycle de vie d'un projet de conception système : c'est aux industriels de choisir un cycle de vie parmi les modèles existants tels que le cycle en cascade ou le cycle en spirale, que [24] liste de façon exhaustive.

Cependant, les industriels affichent généralement l'adoption d'un cycle en V [2] [20], qui favorise une séquence chronologique des processus dans le temps et une répartition du travail entre les différentes parties prenantes (client, entreprise ingénierie système, fournisseurs), comme illustré dans la Figure 4, adaptée d'une proposition faite par [22].

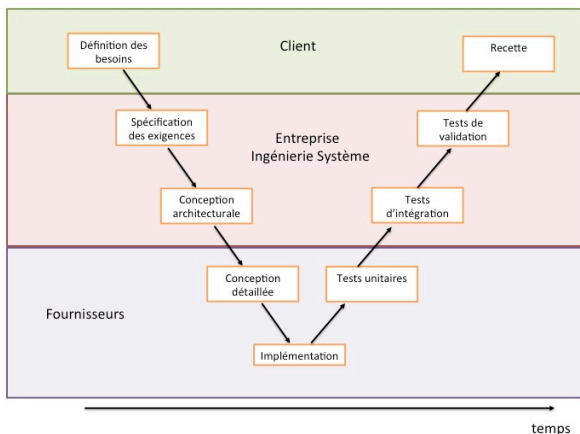


Figure 4: répartition du travail entre parties prenantes selon le cycle en V

D'autre part, les systèmes conçus par les industriels étant de plus en plus complexes, les ingénieurs élaborent et maintiennent un nombre croissant d'exigences et de liens entre exigences-composants-interfaces. Pour les assister dans ces activités, ils utilisent des outils de gestion des exigences, nombreux sur le marché : la société savante internationale en ingénierie système, l'INCOSE, a par exemple répertorié 34 outils différents [10]. On peut citer

les plus connus tels que DOORS édité par IBM, ou CORE édité par Vitech.

L'utilisabilité d'un outil est primordiale dans l'efficacité de l'utilisateur dans ses tâches. Une erreur dans l'ingénierie d'un système peut avoir des conséquences sur la sécurité des opérations du système. De plus, à l'heure où la compétition entre industriels met une forte pression sur les coûts et le temps de développement d'un système, le choix d'un outil efficace pour la réalisation des tâches d'ingénierie des exigences devient stratégique.

ETAT DE L'ART SUR L'UTILISABILITE DES OUTILS D'INGENIERIE DES EXIGENCES

Nous proposons un état de l'art sur les travaux relatifs à la définition et l'évaluation de l'utilisabilité des outils d'ingénierie de exigences. Cet état de l'art est structuré selon trois axes : le cadre d'analyse des outils (framework), la méthode d'analyse utilisée, les résultats de l'analyse en termes d'exigences d'utilisabilité.

Le rapport technique ISO/IEC TR 24766 [14] décrit les capacités attendues des outils d'ingénierie des exigences, et propose de classer ces capacités selon huit catégories : élévation, analyse, spécification, modélisation, vérification et validation, management, traçabilité, et « autres ». Dans la catégorie « autres », nous trouvons deux aspects qui nous intéressent plus particulièrement en IHM : l'accès collaboratif et l'interface graphique. La collaboration sur les exigences est envisagée à travers la gestion des accès aux données : gestion des rôles et des groupes utilisateurs, accès aux données en lecture ou écriture selon les rôles. Les aspects relatifs à l'interface utilisateur sont très vagues : le rapport préconise que les outils fournissent des vues multiples (sans préciser lesquelles), des entrées et des manipulations interactives graphiques des données (sans préciser), une interface web, les fonctions de copier-coller, et des capacités classiques des systèmes d'exploitation tels que le multifenêtrage et l'accès concurrent à plusieurs fonctions. Ce rapport se présente comme une aide à la comparaison et à l'évaluation des outils pour les entreprises dans un processus d'achat, mais ne propose pas lui-même d'évaluation des outils avec ce cadre. L'INCOSE a par contre utilisé ce cadre pour évaluer les outils répertoriés [10], mais en demandant aux éditeurs des outils de faire l'évaluation : chaque outil est (forcément) bien évalué par son propre éditeur.

[17] propose une évaluation statistique de 38 outils utilisés par les praticiens, en utilisant également l'ISO/IEC TR 24766 comme cadre, mais ne retient que l'aspect de l'interface web comme critère, montrant la difficulté de préciser les autres aspects. La méthode est contestable car ce sont également les éditeurs d'outils qui ont évalué leur outil sur la base d'un questionnaire.

[3] propose un cadre d'analyse des visualisations à partir des activités d'ingénierie des exigences et d'un cycle de vie des exigences. Les activités d'ingénierie des

exigences sont très similaires aux catégories proposées par ISO/IEC TR 24766 : élicitation, modélisation et analyse, communication et négociation, vérification et validation. Nous notons cependant deux différences : les activités de négociation et communication sont reconnues comme des activités de premier plan, alors que les activités de management des exigences n'apparaissent plus de façon explicite. Ces activités de management sont transférées dans une phase du cycle de vie des exigences. Ce cycle de vie des exigences est composé de 4 phases : mise en place, élaboration, raffinement, spécification, ces phases se succédant de façon chronologique. En matière de *méthode*, l'évaluation est réalisée par 4 chercheurs en ingénierie des exigences, en utilisant ce cadre croisé activités/phases, et porte sur des visualisations issues de 29 articles de recherche. Ils introduisent également une catégorisation des visualisations : tabulaire, relationnelle, séquentielle, hiérarchique, quantitative/métaphorique. En matière de *résultats*, dans leur effort de catégorisation, les auteurs se rendent compte du recouvrement entre activités et phases, lié notamment à la nature itérative du travail fait sur les exigences. Ils notent une prévalence des visualisations de type relationnelle et hiérarchique, sur les phases de raffinement et de spécification. C'est pour la phase d'élaboration des exigences que la plus grande variété de visualisation est constatée (séquentielle, tabulaire, hiérarchique et relationnelle). En revanche, peu de visualisations sont proposées pour les activités de vérification et validation des exigences. Les auteurs concluent en préconisant le développement de visualisation pour les tâches de vérification et validation, et l'ouverture vers des visualisations quantitatives/métaphoriques, sans formuler d'exigences plus précises. Ils préconisent également d'aller vers une meilleure compréhension des tâches à instrumenter à travers les visualisations, conscients des limites de leur cadre d'évaluation. Nous pouvons noter que l'interaction sur les visualisations n'est pas du tout prise en compte dans leur évaluation.

Au contraire, [21] identifie l'interaction sur les visualisations d'exigences comme un moyen d'exploration, de découverte de connaissances et de raisonnements efficaces. Une analyse des récents travaux sur la visualisation des exigences est proposée par les auteurs, avec également un cadre. Ce *cadre* n'est plus centré sur les activités d'ingénierie des exigences, mais sur un processus de visualisation de données par l'utilisateur : l'utilisateur sélectionne des données, les données correspondent à un « modèle » du système, l'utilisateur spécifie le « modèle », le « modèle » est transformé en visualisation, l'utilisateur interagit sur la visualisation pour comprendre et découvrir. L'interaction sur la visualisation est proposée comme de première importance pour l'utilisateur, alors que la sélection des données et l'édition du modèle sont évaluées comme secondaires. Nous retenons de ce cadre les « questions opérationnelles » identifiées sur les visualisations : des

vues multiples, avec des possibilités de navigation, recherche, filtrage et annotation sur ces vues. En matière de *méthode*, l'évaluation est conduite par les deux auteurs de l'article, puis discutée avec un troisième chercheur, sur 8 outils issus de la littérature académique ayant une approche de « visual analytics » (VA) sur les exigences. Ils notent qualitativement sur une échelle de 4 l'atteinte des questions opérationnelles de leur cadre. *Les résultats* montrent que les aspects visualisation, en termes de vues multiples, navigation entre vues, recherche, filtrage, requêtes et annotation sont mal couverts par les outils proposés. La proposition de vues multiples par les outils est cependant le point le mieux traité. Les aspects liés à la découverte de connaissances, tels que la détection d'anomalies, les explications détaillées, l'aide au raisonnement, sont également mal couverts. Les auteurs ne formalisent pas les résultats sous forme d'exigences à remplir par les visualisations et les interactions, mais restent sur cette notion de « question opérationnelle ».

Dans un deuxième temps, les auteurs reportent une étude de cas sur un projet réel, impliquant 3 utilisateurs, dans laquelle ils demandent aux participants d'utiliser l'outil qu'ils ont conçu, RecVisu+, pour passer en revue et analyser les exigences du projet réel des utilisateurs. Les auteurs n'utilisent plus leur cadre d'évaluation, mais se centrent sur les tâches d'ingénierie des exigences (quelles sont les tâches qui ont besoin de VA? comment VA supportent ces tâches? quels bénéfices peut-on attendre?). En matière de *méthode*, l'évaluation est faite lors de 4 séances, alternant observation et discussion, par 3 utilisateurs-ingénieurs, sur l'outil RecVisu+. *Les résultats* de l'étude sont présentés sous forme de tâches d'ingénierie des exigences intéressantes à instrumenter par du visual analytics, et comment l'outil RecVisu+ fournit ou non ce support. Les auteurs ne formalisent pas ces résultats sous forme d'exigences en tant que telles pour les outils d'ingénierie des exigences, mais plutôt comme des améliorations à apporter à leur outil.

[8] affiche clairement comme objectif de préciser les exigences pour les outils de gestion des exigences (c'est son titre). Les auteurs prennent comme *cadre d'analyse* le standard ISO/IEC 9126 d'évaluation des caractéristiques qualité d'un logiciel [15]. La *méthode* utilisée est basée sur l'expérience des auteurs dans des projets de développement de système. Les nombres de projets, d'années d'expérience ne sont pas précisés. Les outils rencontrés lors de ces expériences ne sont pas précisés non plus. En termes de *résultats*, les auteurs présentent un catalogue d'exigences selon trois acteurs différents, le développeur, l'administrateur du projet, et l'administrateur de l'outil. Bien que les auteurs annoncent que les exigences sont purement fonctionnelles et ne couvrent pas des aspects généraux tels que la « user-friendliness », on peut identifier des exigences d'utilisabilité dans le catalogue. Il existe notamment une partie « vues », dans laquelle les exigences restent floues (« graphical views of the

requirements should be available », « all users must be able to customize the standard views without changing the template »).

Synthèse

Notre état de l'art montre de nombreuses tentatives de définition de cadres d'analyse des outils, qui se révèlent difficiles à utiliser par les auteurs mêmes de ces cadres. Cette difficulté d'utilisation peut s'expliquer par une similitude de ces cadres avec les activités prescrites de l'ingénierie des exigences. Une meilleure compréhension des tâches à instrumenter est d'ailleurs préconisée de façon récurrente [1] [5] [21].

Les méthodes d'analyse des outils sont à base de jugement d'expert [3] [21] [8], ou de questionnaires remplis par les éditeurs des outils [10][17]. Lorsque ces évaluations impliquent des utilisateurs, comme dans [21], les résultats de ces évaluations ne sont pas généralisées à des exigences que doivent remplir les outils, mais restent centrés sur des améliorations de l'outil évalué.

Finalement, les exigences sur l'utilisabilité des outils d'ingénierie des exigences restent mal formulées si on y applique les règles de bonne formulation d'une exigence (**Figure 1**). L'exigence la plus récurrente est de proposer des vues multiples des mêmes données (viewpoint ou views) à l'utilisateur, mais sans les préciser de façon non ambiguë. L'interactivité sur les vues est présentée par [21] comme prometteuse pour la découverte de connaissances par l'ingénieur (détection d'anomalie, explication détaillée, raisonnement par hypothèse ou scénario), mais n'est pas formalisée sous forme d'exigences précises.

ANALYSE DE L'ACTIVITE DE L'INGENIEUR SYSTEME DANS L'INDUSTRIE AERONAUTIQUE

Nous proposons de construire une meilleure compréhension des tâches d'ingénierie des exigences à instrumenter, telles qu'elles sont actuellement réalisées par les industries aéronautiques. Après avoir expliqué la méthode utilisée, nous présentons les résultats de notre analyse de l'activité en caractérisant d'abord les projets et les données produites lors de l'activité sans en révéler les contenus détaillés pour des raisons de confidentialité. Nous présentons ensuite des scénarios d'usage, en formalisant les forces et faiblesses relevées sous forme d'exigences d'utilisabilité.

Méthode

Nous avons choisi comme méthode la réalisation d'enquêtes contextuelles auprès d'ingénieurs, afin de bien identifier les problèmes réels des ingénieurs, comme suggéré par [1]. Notre objectif est d'identifier les tâches réalisées, cerner les problèmes d'utilisabilité de leurs outils, et formuler des exigences plus précises en termes d'interactions, de visualisations et de collaboration.

9 ingénieurs ont été interviewés de janvier à mars 2014, de 4 entreprises différentes, dans l'industrie aéronautique. 6 ingénieurs sont confirmés, avec au moins 10 ans d'expérience, les 3 autres ayant 4 ans d'expérience. Chaque interview a duré 2 heures, et a fait l'objet d'un enregistrement vidéo, sauf pour 3 ingénieurs pour des raisons de confidentialité. Une transcription de chaque interview a été réalisée, en y incluant des captures de vidéo pour préciser les propos. Des données ont été récupérées auprès de 6 ingénieurs, telles que des schémas d'architecture, des documents de spécification, des référentiels d'exigences, des matrices de traçabilité entre exigences et tests, des présentations, des cahiers de tests, des comptes rendus de réunion, des échanges de mails entre ingénieurs. Ces données ont été exploitées lors des transcriptions pour identifier les détails des visualisations et des interactions utilisées lors des interviews.

Caractérisation des projets et des données

Parmi les ingénieurs interviewés, nous avons 5 projets *d'évolution* de systèmes existants. Le système est déjà opérationnel, les composants du système et les interfaces entre ces composants sont connus, il s'agit de faire évoluer le système existant avec un nouveau service. 3 projets sont relatifs à la prise en compte d'évolutions de systèmes bord (IHM cockpit d'un hélicoptère, service d'anti-collision d'un avion de transport, évolution d'une suite avionique d'un turbopropulseur). 2 projets sont relatifs à la prise en compte d'évolutions de système sol (IHM et service de collaboration pour la gestion des départs, service de communication entre actuel et futur systèmes de contrôle du trafic aérien).

Cette situation est assez fréquente pour l'ingénieur système, la durée de vie des systèmes sol et bord étant d'au moins 20 ans. Pendant cette durée, la maintenance corrective et évolutive doit être assurée : la prise en compte des anomalies (ou faits techniques) suit souvent le même processus d'ingénierie système qu'une évolution. La mise en place d'une nouvelle version d'un système est coûteuse, car elle exige la production de tous les documents de preuve de conformité pour la certification.

Parmi les autres projets observés, un projet est relatif à un nouveau système de radio-téléphone pour le contrôle aérien, avec appel d'offre, discussion et choix des fournisseurs. Les 3 autres projets concernent des projets exploratoires, où des travaux de modélisation système sont mis en œuvre en préparation d'affaires futures.

Les documents produits sont des documents de spécification (System SubSystem Specification, Software Requirements Specification), où le nombre d'exigences système va de 100 à 2500 pour une nouvelle fonction. Quand on passe au niveau sous-système, une exigence système est affinée en au moins 3 exigences sous-systèmes. Quand on passe niveau logiciel, chaque exigence sous-système est encore affinée, avec un

facteur plus important s'il s'agit d'IHM. En effet, les exigences peuvent être très détaillées, avec des captures d'écran issues de prototypage à l'appui (*ex : If more than two aircraft are displayed in the ITP traffic list, the 'TOO MANY TRAFFIC' message shall be displayed in amber large font characters on the LSK 2R label line (refer to Figure 68)*). Le système FMS (Flight Management System) d'un avion est défini par 16 000 exigences.

Les scénarios présentés ci-après vont être illustrés par des extraits de ces documents. Nos exigences d'utilisabilité seront intercalées entre crochets dans les scénarios, les exigences seront détaillées dans le tableau de la Figure 8 afin de ne pas surcharger la lecture des scénarios. Les scénarios sont numérotés afin de pouvoir les référencer.

1- Conception exploratoire et partagée de l'architecture entre ingénieur système et fournisseurs des composants

Dans trois entreprises différentes, nous avons constaté que l'ingénieur système adopte une façon de travailler différente de l'activité prescrite que nous avons présentée précédemment. En effet, il ne va pas réfléchir seul à l'allocation des exigences système entre composants, puis rédiger les exigences pour chaque composant, pour enfin les transmettre aux fournisseurs des composants.

A l'inverse, il choisit de lister les problèmes de conception que soulèvent les exigences système à remplir (ou les anomalies à traiter), et travaille avec les fournisseurs des composants lors d'« ateliers » sur ces problèmes de conception. Ce travail débouche sur un panel de solutions techniquement réalisables en termes d'allocation d'exigences, car proposées et discutées avec les fournisseurs. Cela permet ensuite à l'ingénieur système de faire un choix entre ces solutions et de rédiger les exigences d'interfaces entre composants. Les exigences affinées par composants sont initiées par l'ingénieur système et complétées par les fournisseurs.

Cette méthode pragmatique est adoptée pour des raisons d'efficacité. Un des ingénieurs système interviewés, se positionnant au sein de son entreprise comme un garant des processus, tient un discours contradictoire: il regrette que "l'IS n'ait pas le vent en poupe" et que "les gens ne suivent pas les processus". Mais il constate aussi l'efficacité d'un travail préalable à la rédaction des exigences en collaboration avec les fournisseurs des différents composants du système. Un autre ingénieur interviewé, travaillant au niveau sous-système, se réjouit de ce travail avec les ingénieurs système, fustigeant « la forme contre le fond », et indiquant l'importance d'« avoir les bureaux les uns à côté des autres ».

En termes d'exigence d'utilisabilité, nous en déduisons que [le système doit être flexible (« pliant ») vis-à-vis des processus d'ingénierie système] [7], pour permettre l'utilisation des méthodes pragmatiques observées. Cette

exigence est certes de haut niveau, mais nous pouvons l'affiner en exigences détaillées en analysant comment cette méthode de travail est actuellement instrumentée, à travers les scénarios d'usage suivants.

Lister les problèmes de conception en dessinant l'architecture

L'ingénieur système utilise un outil de dessin, tels que Visio pour construire un schéma de l'architecture. Le système d'étude est un système pour le contrôle aérien. Il est opérationnel, les composants du système et les interfaces entre ces composants sont donc connus. Il s'agit de faire évoluer le système existant avec un nouveau service. L'ingénieur système utilise ce schéma pour identifier les problèmes de conception, et cela de façon exhaustive. La construction du schéma lui permet d'avoir une vision d'ensemble, et d'« être sûr de ne rien louper » [le système doit permettre la conception exploratoire][6]. Il préfère utiliser des logiciels de dessin plutôt que des outils de modélisation système car les dessins vont évoluer dans le temps et doivent être facilement modifiables [le système permet de dessiner un schéma d'architecture facilement modifiable]. « Ce sont des hypothèses d'étude, et du fait que c'est un dessin, cela se change de manière très simple ». C'est au moment où il construit le schéma qu'il va discerner/découvrir les problèmes à résoudre (« avec ce genre de schéma, les sujets apparaissent d'eux-mêmes »). En parallèle à la construction du schéma, il va alors rédiger ces problèmes dans un fichier excel, en les structurant par thème/questions/référence vers le document de spécification système (problème d'indirection). Les liens entre les problèmes de conception listés et le schéma ne sont pas explicités, ils sont dans la tête de l'ingénieur. Quand ils le sont (Figure 5), ils sont fait dans un fichier Word, dans lequel le schéma est inséré, et sur lequel on rajoute une flèche et du texte se référant au numéro du problème, créant une indirection [le système permet d'éditer une liste de problèmes de conception, en lien avec le schéma d'architecture et les exigences système].

Partager et discuter les problèmes de conception et l'architecture

Dans un deuxième temps, le schéma d'architecture et la liste des problèmes de conception sont présentés aux fournisseurs des composants du système lors d'une réunion de travail. L'objectif de la réunion pour l'ingénieur système est de présenter son schéma et les problèmes de conception identifiés aux fournisseurs, et voir avec eux s'« ils ont déjà la solution de leur côté pour réaliser la fonction ». Nous n'avons pas assisté à ce genre de réunion, mais selon les sujets, la réunion aborde un par un les problèmes de conception identifiés dans le document, permet d'avoir une compréhension partagée des problèmes et envisager et confronter des pistes de solution réalisables [le système permet de partager la listes des problèmes de conception et le schéma en réunion colocalisée ou à distance]. Les pistes

de solution ne sont pas notées en séance dans les documents, mais précisées a posteriori par messagerie électronique. Lorsqu'on leur demande les résultats de ces réunions, l'ingénieur utilise sa messagerie en filtrant sur le mot « atelier » ou sur la date de la réunion, et prend du temps pour trouver l'information. Il n'y a pas vraiment de compte-rendu formel de ces ateliers : l'ingénieur compte alors essentiellement sur sa mémoire, sa connaissance du système, son expertise « je n'ai pas besoin de compte-rendu pour répondre aux questions techniques ». En revanche, en cas de transfert de l'activité vers un collègue, « ce serait plus difficile » [le système permet aux différents utilisateurs de renseigner leur analyse sur les problèmes de conception].

Les exigences système initiales peuvent être modifiées suite à ces analyses, ainsi que le schéma d'architecture, pour intégrer les pistes envisagées [le système permet de gérer plusieurs versions de schémas d'architecture, de visualiser les différences entre les schémas, de lier les différences aux analyses des problèmes de conception].

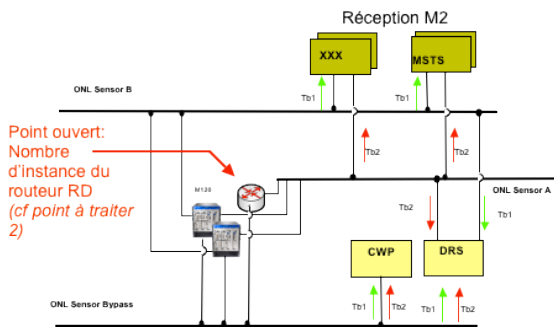


Figure 5: un schéma d'architecture incluant un référence indirecte à un point à traiter

Suivre l'avancement du travail sur les problèmes de conception, l'architecture et les exigences

Certains problèmes de conception sont clos, et l'ingénieur système va commencer à rédiger les exigences pour chacun des composants dans un fichier Word, en indiquant « TBD » (to be defined), que le fournisseur doit compléter. Pour cela, il va intégrer son schéma dans un fichier Word, ajouter au-dessus les numéros des exigences, et créer un paragraphe « exigences » dans lequel on va retrouver les numéros avec le texte associé (Figure 6). [le système permet de créer et modifier une ou plusieurs exigences à partir d'un élément graphique du schéma d'architecture][le système permet d'accéder aux exigences à partir d'un élément graphique du schéma d'architecture].

Certains problèmes ne sont pas clos, et doivent être étudiés par un ou plusieurs fournisseurs. Pour gérer ces problèmes non clos, l'ingénieur utilise un logiciel de gestion de projet, dans lequel il va reporter en tant qu'activités du planning les différentes solutions envisagées avec un temps alloué à chacune des solutions. Cependant, pour expliquer l'avancement du travail, l'ingénieur va revenir sur le schéma d'architecture, le

passer en revue avec son curseur de souris et en verbalisant « ça c'est bon, ça c'est en cours, etc ». [Le système permet de suivre l'avancement des analyses des problèmes de conception à travers le schéma d'architecture].

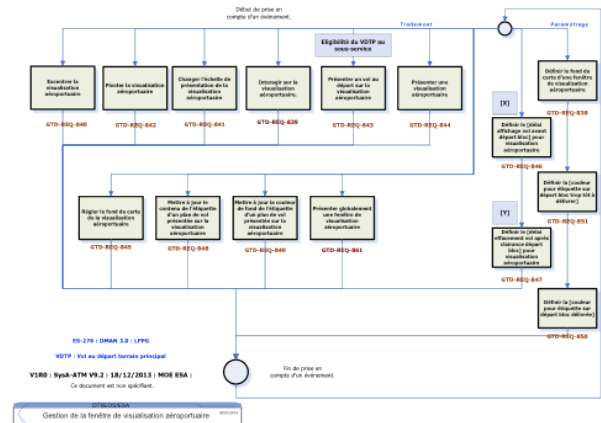


Figure 6: schéma d'architecture fonctionnelle, où les exigences sont en rouge sous les objets graphiques, avec une indirection sur l'énoncé de l'exigence via le numéro

2- Une modélisation fastidieuse et illisible

Trois ingénieurs interviewés utilisent des outils de modélisation système, utilisant des formalismes tels que SysML. Certaines entreprises s'orientent en effet vers ce genre d'outils, afin de contrer la croissance incontrôlée des exigences (« un schéma vaut mieux qu'une tonne de texte »). L'intérêt de ces outils est de pouvoir éditer l'architecture fonctionnelle, préciser des performances et des contraintes sur les fonctions dans un deuxième temps, pour ensuite pouvoir simuler le fonctionnement du système modélisé. La modélisation et la simulation de solutions alternatives permettent alors de comparer les solutions et de faire des choix argumentés.

Nous avons pu constaté lors des interviews que les outils de modélisation ne sont pas utilisés de cette façon. En effet, seule la solution retenue est modélisée, le travail de comparaison des solutions et de décision se faisant par jugement d'expert lors des ateliers entre ingénieurs que nous avons précédemment décrits. La conséquence est que les solutions alternatives ne sont pas enregistrées, ni les discussions et compromis qui ont conduit à cette solution.

Cette situation peut s'expliquer par deux points que nous avons pu observer : les problèmes d'utilisabilité lors de l'édition du modèle, et les problèmes de lisibilité du modèle produit. Par exemple sur le modèle de la figure 7, l'alignement des différents rectangles a été réalisé manuellement par l'ingénieur, un par un. Les rectangles complètement à droite représentent des instances d'un même acteur du système. Si l'ingénieur a oublié une sortie sur les rectangles à droite du modèle, il peut ajouter une sortie sur l'acteur, cela ajoutera la sortie sur tous les rectangles. Mais s'il veut positionner

graphiquement la sortie à un endroit donné sur le côté du rectangle, il faut qu'il réalise cette opération sur chacun des rectangles (« c'est boîte par boîte, il manque des macros »). Il faut de plus qu'il déplace les rectangles un par un, car trop près du bord de la fenêtre [Le système permet de modifier les propriétés graphiques d'un ensemble d'objets]. Si l'ingénieur veut modifier les exigences sur l'instance, il double-clique sur le rectangle, ce qui déclenche l'ouverture d'une fenêtre par dessus le modèle, qui contient 11 onglets (

Figure 7: exemples de visualisation de modèle et de fenêtre de gestion des propriétés) : « on a l'impression que les propriétés sont dans l'ordre inverse de ce dont on a besoin ». [Le système permet de rechercher facilement une propriété du domaine et de modifier cette propriété du domaine pour un ensemble d'objets].

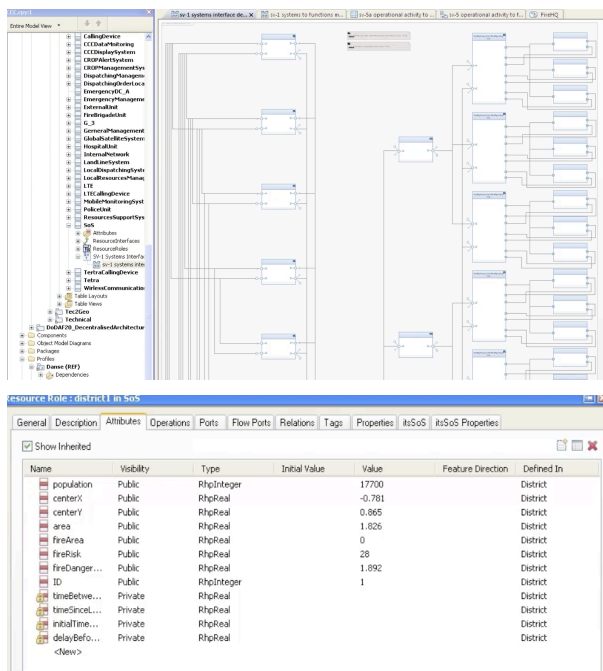


Figure 7: exemples de visualisation de modèle et de fenêtre de gestion des propriétés

Enfin, comme il n'y a pas de noms sur les boîtes (de gauche à droite il s'agit de quartiers d'une ville, de quartier général, de casernes, et de pompiers), le modèle n'est pas facilement compréhensible par une autre personne [Le système permet d'associer une représentation visuelle à un objet]. L'ingénieur reconnaît que le modèle n'est pas lisible pour soi-même quelques semaines après : « le travail n'est pas quotidien, l'outil devrait permettre de se souvenir de ce qu'on a fait, où on en est, comment on fait, quelle est la prochaine étape à réaliser » [Le système permet d'identifier les dernières actions d'édition réalisées sur le schéma d'architecture].

3-Consitution et maintenance du référentiel d'exigences

Certains ingénieurs interviewés sont responsables de la gestion des exigences. "DOORS est super pour trois

choses : maintenir la traçabilité, créer des baselines, permettre l'accès multi-utilisateurs avec la possibilité de verrouiller des zones." En effet, DOORS permet de construire et maintenir le référentiel des exigences pour une version de système. Il permet de générer selon des modèles les documents nécessaires à la certification, notamment les preuves de traçabilité exigées par la norme DO178B [4] : « Traceability between system requirements and software requirements should be provided to enable verification of the complete implementation of the system requirements and give visibility to the derived requirements ».

L'édition et la visualisation des liens dans DOORS soulèvent des problèmes d'utilisabilité. Pour construire la traçabilité sur chaque exigence il y a la possibilité d'insérer des liens (flèche jaune *in* ou flèche rouge *out*) qui renvoient seulement vers d'autres éléments existant déjà dans DOORS. Si ces éléments n'existent pas, l'ingénieur système est obligé de créer d'abord ces éléments, pour ensuite pouvoir les référencer comme lien, lui faisant perdre son contexte [Le système permet de créer un lien à partir d'une exigence en conservant le contexte de l'utilisateur].

Le retour d'information sur une exigence système qui a été dérivée en une ou dix exigences est le même, ce qui donne l'impression que « tout est à plat, il y a un problème de structure ». [Le système permet sans interaction de voir le nombre d'exigences dérivées pour une exigence] [Le système permet d'avoir une vue d'ensemble structurée des exigences]. Pour voir les exigences dérivées, il faut sélectionner la flèche jaune, déclenchant l'affichage de la liste des chemins de répertoire des exigences dans DOORS (ex : /Projet/Referentiel/Evolutions/R347) : il est alors difficile de comprendre le contenu des exigences dérivées [Le système permet d'atteindre l'énoncé des exigences dérivées à partir d'une exigence en une interaction au maximum]. De plus, les exigences dérivées ne sont accessibles qu'une par une à travers la liste [Le système permet d'atteindre l'ensemble des exigences dérivées à partir d'une exigence].

Notre ingénieur système est chargé de la vérification de la cohérence : il a mis en place dans un document Word un glossaire de termes, mais ce glossaire n'est pas stocké dans DOORS. La conséquence est l'utilisation de termes différents dans les exigences pour désigner le même concept (ex : « famille de vols » vs « liste de vols ») [Le système propose à l'utilisateur les termes conformes à un glossaire].

Le problème existe également avec les verbes : « si un nouvel ingénieur arrive sur le projet, il va créer une exigence *interagir avec la timeline de séquençement* alors qu'il existe déjà une exigence *manipuler la timeline de séquençement* ». [Le système permet à l'utilisateur de trouver les exigences similaires sémantiquement en mettant en valeur les similitudes]. Nous précisons « à l'utilisateur » dans cette exigence, afin de privilégier des

Enoncé	Justification
Le système doit être flexible vis-à-vis des processus d'ingénierie système : <ul style="list-style-type: none"> • éditer une liste de problèmes de conception, en lien avec le schéma d'architecture et les exigences système non finalisées ; • partager la liste des problèmes de conception et le schéma d'architecture en réunion colocalisée ou à distance (cscw) ; • éditer l'analyse sur les problèmes de conception par plusieurs utilisateurs (cscw) ; • visualiser les différences entre plusieurs versions de schémas d'architecture ; • lier les différences aux analyses des problèmes de conception. 	scénario 1
Le système permet de dessiner un schéma d'architecture <i>facilement</i> modifiable : <ul style="list-style-type: none"> • modifier les propriétés graphiques d'un ensemble d'objets ; • rechercher facilement une propriété du domaine d'un objet ; • modifier une propriété du domaine pour un ensemble d'objets ; • associer une représentation visuelle à un objet. 	scénarios 1 et 2
Le système permet de suivre l'avancement du travail à partir du schéma d'architecture : <ul style="list-style-type: none"> • voir les problèmes de conception clos, avec leur analyse ; • voir les problèmes de conception en cours d'analyse, avec une date de fin estimée ; • voir les dernières actions d'édition réalisées. 	scénarios 1 et 2
Le système permet une navigation à travers le schéma d'architecture : <ul style="list-style-type: none"> • créer une exigence à partir d'un élément graphique du schéma ; • accéder facilement aux exigences à partir d'un élément graphique du schéma. 	scénario 1 [21]
Le système permet de gérer facilement des liens entre exigences : <ul style="list-style-type: none"> • créer un lien à partir d'une exigence en conservant le contexte de l'utilisateur ; • voir sans interaction le nombre d'exigences dérivées pour une exigence ; • atteindre à partir d'une exigence l'énoncé des exigences dérivées en une interaction au maximum . 	scénario 3
Le système permet de naviguer facilement dans un ensemble d'exigences textuelles : <ul style="list-style-type: none"> • avoir une vue d'ensemble structurée des exigences ; • atteindre à partir d'une exigence l'ensemble des exigences dérivées ; • atteindre à partir d'une exigence dérivée l'exigence de plus haut niveau. 	scénario 3 [21]
Le système permet à l'utilisateur de vérifier la cohérence d'exigences textuelles : <ul style="list-style-type: none"> • propose à l'utilisateur les termes conformes à un glossaire ; • aide l'utilisateur à trouver les exigences similaires sémantiquement en mettant en valeur les similitudes. 	scénario 3

Figure 8: synthèse des exigences d'utilisabilité

solutions où l'utilisateur reste actif dans la détection et la correction des exigences, les solutions automatiques de correction et de vérification d'ontologie étant nombreuses mais pas nécessairement utilisables [1].

SYNTHESE DES EXIGENCES

Le Figure 8 est une synthèse des exigences d'utilisabilité identifiées dans les travaux antérieurs et les scénarios. Nous avons affiné ces exigences en exigences détaillées quand nous avons assez d'éléments pour le faire : comme dans tout processus d'ingénierie système, nos exigences seront affinées quand nous rentrerons dans la conception de solutions. Nous justifions la pertinence des exigences grâce à nos scénarios d'usage issus des enquêtes contextuelles réalisées.

CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons élicité les exigences d'utilisabilité pour les outils d'ingénierie des exigences.

A la différence des travaux existants : (1) nous nous sommes appuyés sur des enquêtes contextuelles de l'activité réelle des ingénieurs système (4 entreprises, 9 interviews retranscrites, des projets de nature différente) ; (2) nous avons formalisé ces exigences en respectant la norme ISO [13], ce qui les rend plus utilisables pour la communauté.

L'utilisabilité d'un outil concerne l'efficacité, l'efficience et la satisfaction de l'utilisateur dans un contexte d'utilisation spécifié [16]. Dans un environnement concurrentiel entre industriels, améliorer l'utilisabilité des outils d'ingénierie des exigences peut dans un premier temps permettre de gagner du temps sur la maintenance corrective et évolutive des systèmes. Dans un deuxième temps, cela peut contribuer à la maîtrise de la complexité des systèmes dans le temps.

En termes de perspectives, nous prévoyons d'utiliser le travail réalisé sur les exigences pour concevoir des outils

d'ingénierie des exigences. Les concepts issus de la visualisation de grande quantité de données et des collecticiels seront particulièrement explorés, notre analyse de l'activité montrant que les activités d'ingénierie des exigences sont fortement collaboratives et génèrent des milliers d'exigences.

REMERCIEMENTS

Nous remercions les entreprises et les ingénieurs qui ont accepté d'être interviewés et observés dans leur environnement de travail.

BIBLIOGRAPHIE

- Betty H. C. Cheng and Joanne M. Atlee.. Research Directions in Requirements Engineering. In Proc. *Future of Software Engineering (FOSE '07)*. IEEE Computer Society, (2007), pp 285-303.
- Boehm B. "Some Future Trends and Implications for Systems and Software Engineering Processes." *Systems Engineering* 9, pp 1-19(2006)
- Cooper, J.R.; Lee, Seok-Won; Gandhi, R.A.; Gotel, O., "Requirements Engineering Visualization: A Survey on the State-of-the-Art", In Proc. Of *Fourth International Workshop Requirements Engineering Visualization (REV)*, (2009), pp.46-55.
- DO-178B/ED-12B Software considerations in airborne systems and equipemnt certification, Requirements and Technical Concepts for Aviation , RTCA SC-167 / EUROCAE WG-12.
- Gotel, O.; Mader, P., "How to Select a Requirements Management Tool: Initial Steps," In Proc. Of *17th IEEE International Requirements Engineering Conference, RE '09*, IEEE (2009), pp.365-367.
- Green,T.R.G.,Cognitive dimensions of notations, *People & Computers V*, 1989, Cambridge Univ. Press, 443-460.
- Harris J. and Henderson A. 1999. A better mythology for system design. In *Proc. of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 88-95.
- Hoffmann, M.; Kuhn, N.; Weber, M.; Bittner, M., "Requirements for requirements management tools, In *Proc. of the 12th IEEE International " Requirements Engineering Conference, 2004 »*, pp.301,308,
- Hughes T.P., *Rescuing Prometheus : four monumental projects that change the world*, New York: Pantheon, 1998.
- INCOSE, Requirements Management Tools Survey, <http://www.incose.org/ProductsPubs/products/rmsurvey.aspx>
- ISO IEC 15288-2008. System engineering and software engineering – System life cycle processes. IEEE (2008).
- ISO/IEC 26702 Standard for Systems Engineering - Application and Management of the Systems Engineering Process," (2007).
- ISO/IEC/IEEE 29148:2011Systems and software engineering -- Life cycle processes --Requirements engineering", (2011).
- ISO/IEC TR 24766:2009, Information Technology – Systems and Software Engineering – Guide for Requirements Engineering tool Capabilities, (2009).
- ISO/IEC 9126, Information Technology – Software Product Evaluation – Quality Characteristics and guidelines for their use, (1991).
- ISO 9241-11 Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation - partie 11 : Lignes directrices concernant l'utilisabilité, (1998).
- Juan M. Carrillo de Gea, Joaquín Nicolás, José L. Fernández Alemán, Ambrosio Toval, Christof Ebert, Aurora Vizcaíno, Requirements engineering tools: Capabilities, survey and assessment, *Information and Software Technology*, Volume 54, (2012), pp 1142-1157
- Karlson A., Benderson B. & Contreras-Vidal J. *Understanding one handed use of mobile devices*. In *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*. IGI Global(2008), pp 86–101.
- Klaus Pohl. 1994. The three dimensions of requirements engineering: a framework and its applications. *Inf. Syst.* 19, 3 (1994), pp 243-258.
- Luzeaux D., Ruault R., Wippler J-L. Maitrise de l'ingénierie des systèmes complexes et des systèmes de systèmes, étude de cas. Hermès (2011).
- Nan Niu; Reddivari, S.; Zhangji Chen, "Keeping requirements on track via visual analytics,In *Proc. Of 21st IEEE International " Requirements Engineering Conference (RE)*, (2013), pp.205-214.
- Sage, A.P., "Systems engineering education," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions* vol.30, no.2 (2000), pp.164-174.
- Sire, S., Chatty, S., Gaspard-Boulinc, H., Colin, F.-R. How can groupware preserve our coordination skills? Designing for direct collaboration. In Proc. of *Interact 99, IFIP* (1999), pp. 304--312.
- Thayer Richard H., Bailin Sidney C., and Dorfman M... *Software Requirements Engineerings, 2nd Edition* (2nd ed.). IEEE Computer Society Press (1997).

Résilience des systèmes interactifs: contribution par une architecture tolérante aux fautes

Camille Fayollas^{2,3}, Philippe Palanque³, Jean-Charles Fabre^{2,4}, David Navarre³,
Eric Barboni³, Martin Cronel³ & Yannick Deleris¹

¹AIRBUS Operations, 316 Route de Bayonne, 31060, Toulouse, France
Yannick.Deleris@airbus.com

²CNRS, LAAS, 7 Av. Du Colonel Roche, F-31400 Toulouse, France
Jean-Charles.Fabre@laas.fr

³ICS-IRIT, University of Toulouse, 118 Route de Narbonne, F-31062, Toulouse, France
{barboni, cronel, fayollas, navarre, palanque}@irit.fr

⁴Univ de Toulouse, INP, LAAS, F-31400 Toulouse, France

RESUME

La recherche pour améliorer la fiabilité des systèmes interactifs s'est, pour l'instant, principalement dirigée vers la prévention d'introduction de fautes par la suppression des bugs lors du développement. Cependant, la complexité des systèmes interactifs est telle que, quels que soient ces efforts, des défaillances apparaissent lors de leur utilisation. Les causes de telles défaillances peuvent être dues à des pannes matérielles intermittentes ou (quand ces systèmes sont utilisés en haute altitude (avions, véhicules spatiaux ...) être déclenchées par des fautes naturelles dues aux particules alpha dans les processeurs ou aux neutrons issus des radiations cosmiques. Cet article propose l'identification exhaustive des problématiques se posant pour améliorer la fiabilité des systèmes interactifs en prenant en compte à la fois les erreurs faites dans les phases de développement et les fautes qui peuvent se produire lors de leur utilisation. Du fait qu'aucune recherche n'a été menée à ce jour pour détecter et éliminer les fautes naturelles, l'article propose aussi une architecture logicielle pour intégrer des mécanismes de tolérance aux fautes dans les systèmes interactifs. L'article présente en particulier comment cette architecture couvre les différents composants d'un système interactif incluant les widgets, le gestionnaire de fenêtres et l'application interactive. Ces concepts sont mis en œuvre sur une application interactive de cockpits d'avions civils.

Mots Clés

Tolérance aux fautes, Architecture logicielle, Résilience, Systèmes interactifs critiques.

ACM Classification Keywords

D.2.2 [Software] Design Tools and Techniques -

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670462>

Computer-aided software engineering (CASE), H.5.2 [Information Interfaces and Presentation]: User Interfaces - Interaction styles

INTRODUCTION

Un système critique est un système où le coût d'une potentielle défaillance serait largement supérieur au coût de développement. Il pourrait ainsi aller jusqu'à engendrer des blessures ou des pertes de vies humaines [15] (ils sont alors appelés « safety-critical » en anglais). Les systèmes interactifs critiques sont maintenant présents dans la plupart des stations de contrôle et commande comme par exemple dans les segments sol des satellites, les cockpits d'avions civils ou militaires, les postes de contrôle aérien etc... La complexité et la quantité de données manipulées, le nombre de systèmes à contrôler et la grande diversité des commandes à déclencher dans des périodes de temps limité ont poussé l'introduction de techniques d'interaction évoluées dans la plupart de ces systèmes.

Construire des systèmes interactifs fiables est une tâche fastidieuse en raison de leur nature très spécifique. Le comportement de ces systèmes réactifs est piloté par des événements. Comme ces événements sont déclenchés par des opérateurs humains, ces systèmes réactifs doivent réagir à des événements inattendus. En sortie, des informations (telles que l'état actuel du système) doivent être présentées à l'opérateur de manière à ce qu'elles puissent être perçues et interprétées correctement. Enfin, les systèmes interactifs exigent d'aborder les aspects matériels et logiciels ensemble (par exemple les dispositifs d'entrée et de sortie ainsi que leurs drivers).

Dans le domaine de la tolérance aux fautes, les expériences ont démontré (e.g. [35]) que les défaillances logicielles pouvaient survenir même si le développement du système avait été extrêmement rigoureux. L'une des nombreuses sources de défaillance est appelée faute naturelle [4]; elles sont déclenchées par les particules alpha ou encore par des neutrons provenant des rayonnements cosmiques et impactent les puces. Les systèmes déployés en haute atmosphère ainsi que ceux déployés dans l'espace (e.g. vol spatiaux habités [25])

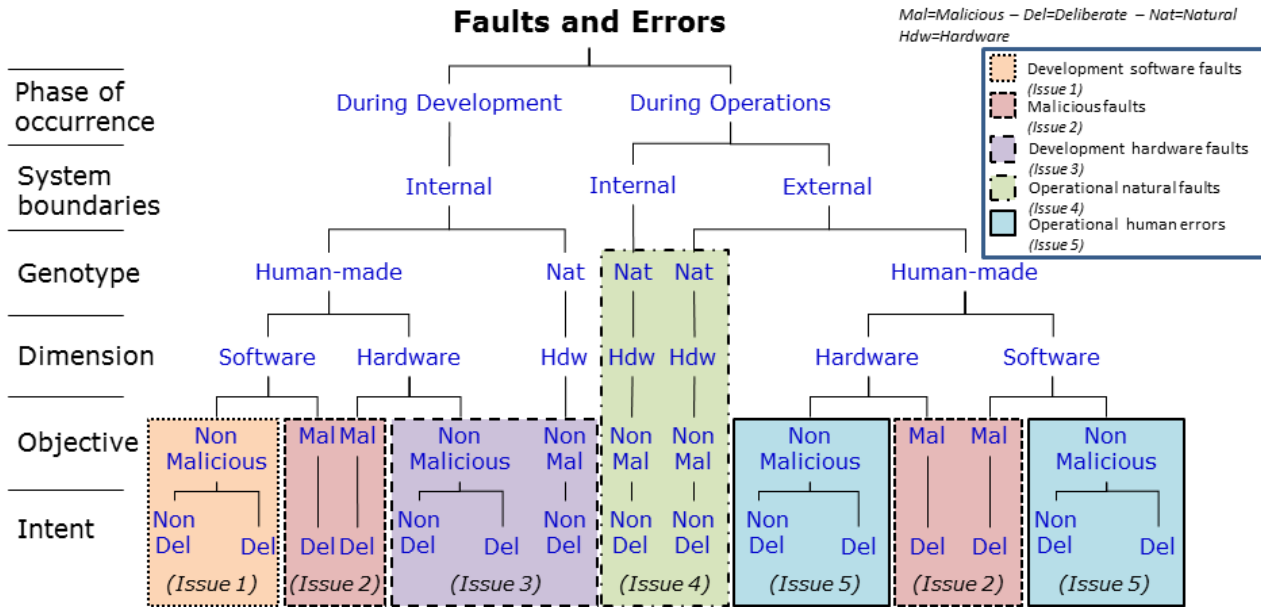


Figure 1. Typologie des fautes des systèmes informatiques (adaptée de [4]) et problématiques associées à leur fiabilité

ont les plus grandes probabilités de défaillance [46] (e.g. avions). Bien que des moyens existent pour se prémunir contre la plupart de ces fautes, elles démontrent le besoin d'aller plus loin que les techniques habituelles de prévention d'occurrences de fautes durant le développement du système (généralement l'utilisation de techniques formelles et la vérification de propriétés) mais aussi celui d'identifier toutes les menaces pouvant impacter les systèmes interactifs.

Cet article traite, dans la première section, de l'identification d'une liste de problématiques qui se posent pour améliorer la fiabilité des systèmes interactifs critiques. La deuxième section traite de l'identification de plusieurs solutions existantes pour répondre à ces problématiques. Une partie de ces problématiques a été en effet étudiée précédemment par la communauté de recherche sur les systèmes interactifs comme par exemple les erreurs provenant des opérateurs. La troisième section porte sur une des questions qui a rarement été étudiée dans ce domaine de recherche : les fautes naturelles qui se produisent indépendamment de l'effort déployé au cours des phases de développement et propose une architecture logicielle tolérante aux fautes pour remédier à ces défauts. Afin d'illustrer les concepts versity Press, 1990. applicabilité, la quatrième section du document présente leur utilisation dans le cadre de l'application de Flight Control Unit présente dans les cockpits modernes interactifs d'avions civils. La cinquième section permet de positionner l'approche proposée dans cet article par rapport aux contributions précédentes des auteurs. Enfin, la dernière section conclut le papier et présente les perspectives de ce travail par rapport aux problématiques présentées dans les premières sections.

PROBLEMATIQUES SOULEVEES PAR LA FIABILISATION DES SYSTEMES INTERACTIFS

Typologie des fautes

Pour s'assurer qu'un système informatique soit fiable [4], c'est-à-dire qu'il soit protégé de tout type de défaillances afin que celles-ci ne puissent pas apparaître ou que leur apparition n'ait pas de conséquence catastrophique sur le système et son environnement, le concepteur de système doit prendre en compte tout type de fautes pouvant impacter le système et déclencher ces défaillances. Dans le but de classier intégralement les fautes pouvant affecter un système informatique, Avizienis et al. ont défini dans [4] une typologie. Cette typologie a permis d'identifier 31 classes de fautes élémentaires. La figure 1 présente une vue simplifiée de cette typologie et rend explicite deux grandes catégories, visibles en haut de la figure : i) les fautes apparaissant pendant le développement (ce qui inclut les mauvaises conceptions et les erreurs de développement) ou ii) les fautes apparaissant pendant le fonctionnement du système (à droite de la figure), ce qui inclut les erreurs comme les ratés, les lapsus et les fautes définis dans [42].

Nous proposons l'organisation des feuilles de la taxonomie en 5 différents groupes amenant chacun une problématique différente :

- *Fautes de développement logiciel (problématique 1)* : introduites involontairement durant le développement du système.
- *Fautes malveillantes (problématique 2)* : introduites délibérément pour provoquer le dysfonctionnement du système (prise de contrôle extérieur, déni de service inopiné, crash du système).

- *Fautes de développement matériel (problématique 3)* : ayant une cause naturelle ou humaine et impactant le matériel durant sa conception.
- *Fautes naturelles en opération (problématique 4)* : causées par un phénomène naturel. Elles affectent le matériel et surviennent pendant le fonctionnement du système, elles sont donc susceptibles d'affecter aussi le logiciel.
- *Erreurs humaines en opération (problématique 5)* : résultantes d'une action humaine pendant le fonctionnement du système. Elles peuvent être matérielles et logicielles et peuvent être délibérées ou non. Le lien entre la typologie et les erreurs humaines dites « classiques » définies dans [42] peut être fait en associant les fautes délibérées aux erreurs ou violations [40] et les non-délibérées avec les ratés et les lapsus.

Approches existantes de prise en compte des fautes

Les études précédentes en sûreté de fonctionnement ont mené à l'identification de 4 moyens d'améliorer la fiabilité d'un système [4] et [15]:

- *La prévention des fautes*: permet d'éviter autant que possible l'introduction de fautes pendant le développement du système. On emploie généralement des techniques de développement rigoureuses, de la formalisation, de la modélisation pour prouver certaines propriétés, ...
- *La suppression de fautes* : permet de réduire le nombre de fautes pouvant survenir: i) pendant le développement du système, généralement en utilisant des techniques de test et d'injection de fautes ou ii) pendant la phase d'exploitation du système en faisant par exemple de la maintenance corrective.
- *La tolérance aux fautes*: permet d'éviter la défaillance de service en présence de fautes via leur détection et leur recouvrement. La détection permet d'identifier la présence de fautes, leur type et leur source. Le recouvrement est généralement réalisé en ajoutant de la redondance ou de la diversification d'un logiciel et permet d'éviter la défaillance.
- *Prévision des fautes* : consiste à estimer le nombre, l'incidence et les conséquences probables de fautes, généralement en dressant une évaluation statistique de la fréquence et de l'impact des fautes.

APPROCHES DE FIABILISATION DES SYSTEMES INTERACTIFS

Comme vu dans la section précédente, la fiabilisation d'un système informatique peut être réalisée si et seulement si toutes les problématiques présentées ci-dessus sont couvertes. La suite de cette section présente les différentes approches existantes pour les traiter tout en prenant en compte les spécificités des systèmes interactifs.

Problématique 1 : Prise en compte des fautes de développement logiciel

Du fait des spécificités des systèmes interactifs [51], les approches standard d'ingénierie des logiciels ne peuvent pas être réutilisées telles quelles pour rendre les systèmes interactifs fiables. De ce fait, dans la communauté de l'ingénierie des systèmes interactifs, de nombreux travaux de recherche ont été réalisés pour raffiner et étendre ces approches. Ces travaux incluent les architectures logicielles [8], les techniques de description formelle et de vérification [33, 24, 9, 16, 31, 14], ou le test [12, 10, 37]. La plupart de ces travaux s'est concentrée sur la suppression des fautes durant le développement du logiciel.

Dans le domaine des cockpits interactifs (que nous présentons dans l'étude de cas), Barboni et al. ont proposé dans [5] de telles méthodes pour décrire de manière complète et non ambiguë les widgets, les gestionnaires de fenêtres et les contrôleurs de dialogue ; couvrant ainsi la partie logicielle des systèmes interactifs. Ceci est également en accord avec le standard de développement DO 178B/C qui recommande l'utilisation de techniques de description formelle [44].

Problématique 2 : Prise en compte des fautes malveillantes

Cet aspect a été très étudié dans le domaine des interactions homme-machine, et a conduit à la création d'un symposium sur la protection de la vie privée appelé SOUPS (Symposium on Usable Privacy and Security) ; où sont discutées les questions d'utilisabilité des systèmes d'authentification [52]. Les questions d'intégrité et de protection des données sont laissées aux domaines liés à la sécurité informatique [43].

Cependant, les problèmes de sécurité informatique spécifiquement liés aux systèmes interactifs critiques n'ont pas encore été traités. De plus, ils commencent tout juste à être abordés dans le domaine des systèmes embarqués, comme présenté dans [18] pour les systèmes embarqués avioniques.

Problématique 3 : Prise en compte des fautes de développement matériel

Les composants matériels utilisés pour les systèmes interactifs sont assez similaires à ceux utilisés dans les systèmes informatiques conventionnels. La prise en compte des fautes de développement matériel peut donc être traitée avec des processus de développement usuels, comme ceux décrits dans le standard DO-254 [19], qui fournit un guide de développement des composants électroniques embarqués.

Cependant, dans le cas d'un système interactif, on doit prendre en compte la fiabilité de l'interaction en plus de la fiabilité du système informatique de base. Par exemple, l'introduction d'une interaction multi-touch dans un cockpit ne pourra être réalisée que s'il a été prouvé que l'écran possède une faible probabilité de défaillance (e.g. moins de 10^{-9} fautes par heure de vol).

Le concepteur du système interactif pourra néanmoins augmenter la fiabilité d'un système interactif en adaptant l'interface ou le logiciel sous-jacent. Par exemple, une interaction tactile peut être fiabilisée en remplaçant un geste linéaire horizontal par un geste oblique afin de ne pas dépendre uniquement d'une ligne de la grille tactile.

Problématique 4 : Prise en compte des fautes naturelles

Comme expliqué dans l'introduction, la prise en compte des fautes naturelles n'a pratiquement pas été étudiée dans le domaine de l'interaction homme-machine et peu de contributions sont disponibles. La proposition de [32] présente la reconfiguration dynamique de la technique d'interaction ou la possibilité de la réorganisation des écrans quand les défaillances matérielles le nécessitent.

Cependant, cette solution n'est pas suffisante et ces aspects ne peuvent être ignorés lorsque l'on traite de la fiabilité de systèmes interactifs critiques. Ceci est particulièrement vérifié pour les systèmes interactifs critiques tels que ceux présents dans les systèmes déployés dans la haute atmosphère (avions) ou dans l'espace (vol spatiaux habités) car une plus forte probabilité d'occurrence de fautes concerne ces systèmes [46,25]. Ces fautes naturelles démontrent la nécessité d'aller plus loin que la prévention classique des fautes durant le développement en prenant en compte les fautes pouvant apparaître en opération.

Problématique 5 : Prise en compte des erreurs humaines

Les erreurs humaines sont étudiées depuis de nombreuses années et peuvent être prévenues ou tolérées [17] et de nombreux moyens ont d'ores et déjà été développés pour la prise en compte de cette problématique. Un premier moyen est de réaliser des barrières [26], qu'elles soient logicielles (e.g. un bouton de confirmation) ou matérielles (e.g. un détrompeur). Bien entendu, les connaissances acquises de l'étude des accidents et des tentatives de violations de barrières doivent être réutilisées pour prévenir la répétition des accidents [7]. Ceci peut-être effectué, par exemple, en incluant des barrières sociotechniques qui pourront être modélisées par des techniques de description formelle afin de prévenir les fautes de développement [6]. Les erreurs humaines peuvent aussi être évitées par la formation spécialisée comme décrit dans [30] ainsi que par un meilleur design de l'aide contextuelle [38] et des manuels d'utilisation [13]. Enfin, un autre moyen de prévenir ces erreurs est d'étudier le comportement humain pour modifier le design des systèmes [41], [36] ou en proposer de meilleurs [49].

Remarques conclusives sur les approches de fiabilisation des systèmes interactifs

La typologie présentée nous permet d'identifier toutes les fautes pouvant affecter un système interactif critique. Elle permet le groupement de certains types de fautes et met donc en exergue les problématiques spécifiques à la

fiabilisation des systèmes interactifs critiques. Dans cette section, nous avons présenté les différentes solutions existantes relatives aux problématiques identifiées. Bien que certaines d'entre-elles aient été largement étudiées, d'autres, en revanche, l'ont été beaucoup moins et nécessitent d'être traitées afin de permettre le développement de systèmes interactifs fiables :

- *Problématique 2* : Les fautes provenant d'attaques malveillantes,
- *Problématique 3* : Les fautes matérielles de développement,
- *Problématique 4* : Les fautes naturelles en opération.

Dans le domaine de la sûreté de fonctionnement, la problématique 2 est principalement résolue par confinement, i.e. complète isolation des systèmes entre eux et plus particulièrement, isolation du reste du monde. Cette solution peut s'avérer problématique compte tenu des maintenances à effectuer qui requerraient alors un accès physique au système. De plus, les communications sont de plus en plus sans fil et ouvrent naturellement les systèmes sur l'extérieur, éliminant ainsi le confinement. Pour ces raisons, la problématique 2 reste à étudier prioritairement. La problématique 3 dépend principalement des constructeurs et du domaine des composants. La problématique 4 a, quant à elle, un fort potentiel d'impacter les systèmes interactifs en opération et nous y consacrerons donc le reste de cet article.

UNE ARCHITECTURE LOGICIELLE POUR DES APPLICATIONS INTERACTIVES TOLERANTES AUX FAUTES

Cette section présente une solution permettant de construire des applications interactives tolérantes aux fautes. Elle propose ainsi une solution à la problématique des fautes naturelles en opération (problématique 4). Dans ce but, nous proposons d'appliquer, en les adaptant, des architectures de tolérance aux fautes aux différents éléments d'une application interactive en se basant sur le modèle ARCH introduit ci-dessous (cf. Figure 2).

Une architecture générique pour les systèmes interactifs

Dans le cadre de cet article, nous nous intéressons particulièrement aux techniques d'interaction classiques (appelées WIMP) pour lesquelles l'affichage et la manipulation se font au travers d'un ensemble prédéfini de widgets. Même si de nombreuses techniques d'interaction bien plus sophistiquées ont déjà été proposées par les chercheurs et industriels du domaine des interfaces homme-machine, la manipulation indirecte reste la plus adaptée aux systèmes critiques car elle correspond aux standards des milieux critiques, comme par exemple le standard ARINC 661 qui définit les interactions dans les cockpits d'avions civils [3]. De plus, les techniques d'interaction indirectes ont déjà été étudiées et standardisées par IBM en 1989 [27].

De telles standardisations présentent de nombreux avantages améliorant significativement la fiabilité (les composants sont réutilisés et donc très largement testés) ; les coûts de développement (les composants sont développés par des entités tierces et sont donc utilisés tel quel par de nombreux clients) ainsi que les temps de développement (les développeurs se concentrent alors sur l'assemblage des composants et non sur leur design).

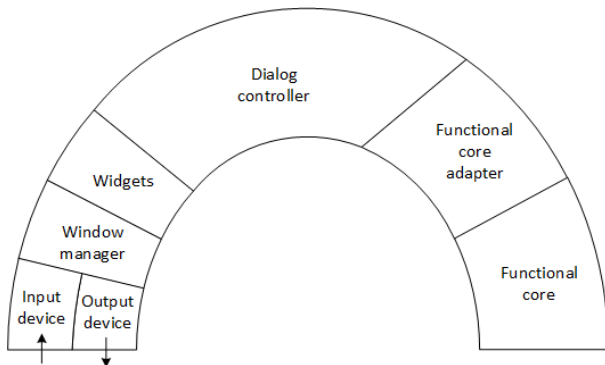


Figure 2. Architecture générique pour les systèmes interactifs adaptée du modèle ARCH [8]¹

La Figure 2 présente une version modifiée de l'architecture logicielle ARCH présentée dans [8]. Cette architecture décrit l'organisation fonctionnelle de la majorité des systèmes interactifs. Elle est utilisée pour représenter de manière exhaustive la variété d'éléments pour lesquels des mécanismes de sûreté de fonctionnement ont été ajoutés :

- *Les périphériques d'entrée et de sortie*: ce sont généralement des claviers, souris et écran... mais ils peuvent être plus complexes (e.g. périphérique combiné comme le KCCU (Keyboard Cursor Control Unit) dans les cockpits). Ils permettent l'interaction physique entre l'humain et la machine.
- *Le gestionnaire de fenêtres* : il encapsule les drivers des périphériques et gère le lien entre les périphériques et le reste de l'application. Par exemple, il est responsable de l'affichage du curseur, de l'identification du widget ciblé par l'action de l'utilisateur (picking), la transmission des événements utilisateurs vers celui-ci ainsi que du rendu graphique de l'information sur le périphérique de sortie.
- *Les widgets*: ce sont les composants de base de l'interaction. Ils sont représentés dans une boîte séparée sur la Figure 2 car nous les considérons comme des composants logiciels indépendants requérant ainsi un mécanisme de tolérance aux fautes dédié comme celui introduit dans [48].

- *Le contrôleur de dialogue*: il décrit les états de l'application ainsi que son comportement.
- *L'adaptateur de code fonctionnel et le noyau fonctionnel* ils encapsulent les fonctions non-interactives du système. Pour cela, nous ne nous concentrerons pas sur ces composants bien qu'ils apparaîtront dans l'étude de cas.

La section suivante est une courte introduction aux architectures tolérantes aux fautes utilisées dans le domaine de la sûreté de fonctionnement. Nous présenterons ensuite les hypothèses sous-jacentes au travail présenté ainsi qu'une description des fautes que nous cherchons à éviter.

Travaux connexes sur les architectures tolérantes aux fautes

Dans le domaine de la sûreté de fonctionnement, de nombreuses approches ont été étudiées pour introduire des mécanismes de tolérance aux fautes au sein des systèmes informatiques. Les techniques les plus utilisées sont : les composants auto-testables [29, 53, 50], la programmation en N-Version [54] et la programmation N-auto-testable [29] (qui est un hybride entre la programmation N-Version et les composants auto-testables). L'architecture présentée dans ce papier est basée sur les composants auto-testables comme de nombreuses stratégies de sûreté de fonctionnement en dépendent. En effet, la couverture de détection d'erreurs d'un composant auto-testable est très bonne et rend les stratégies de réplication possibles afin de pouvoir assurer une continuité de service. C'est d'ailleurs dans ce but que nous proposons également une architecture N-auto-testable qui permet le recouvrement de fautes.

Selon [29], une approche par composants auto-testables consiste à ajouter de la redondance interne à un programme pour qu'il puisse vérifier sa propre dynamique en exécution. Elle est aussi appelée COM-MON [50] ; COM correspondant à COMmande et MON à MONiteur. Cette redondance peut être implantée de 2 manières différentes :

1. Un programme (le COM) auquel on ajoute un programme vérifiant son exécution (le MON, habituellement appelé checker dans ce cas).
2. Un programme (le COM – aussi appelé fonctionnel dans ce cas) auquel on adjoint une variante de ce programme (habituellement appelée contrôleur) et un mécanisme de comparaison ; le MON est alors composé de l'ensemble contrôleur/comparateur.

Dans les deux cas, les fautes naturelles sont détectées par les mécanismes de redondance présents dans le MON si les deux composants logiciels sont exécutés de manière ségréguée, pour éviter les fautes engendrées par les dégâts collatéraux (une faute dans le COM déclenchant une faute dans le MON) et les fautes de mode commun (une faute naturelle impactant à la fois le COM et le MON). L'approche par composants auto-testables a donc la capacité d'assurer la détection des fautes et la

¹ De même qu'en ce qui concerne les aspects tolérance aux fautes, nous avons décidé de garder le vocabulaire anglais pour les figures de l'article dans le but d'éviter les confusions liées à la traduction.

notification d'erreurs lorsque les fautes sont détectées (e.g. quand il y a une différence de comportement entre le COM et le MON).

Le recouvrement de fautes n'est pas possible avec cette approche et il faut donc l'étendre. Ceci peut être fait en utilisant les concepts des composants N-auto-testables qui peuvent être considérés comme de la redondance active, comme expliqué dans la section 5.2 de [11].

Hypothèses d'étude et principales défaillances prises en compte

Cette étude porte sur la fiabilisation du logiciel par rapport aux fautes naturelles, nous considérerons donc que les erreurs humaines sont hors du champ d'application. Cette hypothèse est très forte mais viable puisque l'on peut considérer que les erreurs humaines n'influencent pas les fautes naturelles du système et vice-versa. La contribution se concentre donc sur les défaillances fonctionnelles des trois principales briques de l'architecture logicielle (ARCH) présentée à la Figure 2 i.e. l'application (le contrôleur de dialogue), les widgets et le gestionnaire de fenêtres. Les périphériques d'entrée et sortie ainsi que le noyau fonctionnel sont considérés comme étant hors du champ d'application de cet article, ils sont traités dans d'autres travaux connexes comme [23].

L'architecture que nous proposons permet de s'assurer que le système interactif traite correctement les événements d'entrée venant de l'opérateur, et effectue correctement le rendu des paramètres reçus du noyau fonctionnel. Pour rentrer dans les détails, nous cherchons à gérer les trois défaillances fonctionnelles possibles qui pourraient mener à une catastrophe (e.g. un crash d'avion) :

- **Affichage erroné**: affichage erroné des données reçues du noyau fonctionnel (e.g. un widget reçoit 10,11 et affiche 23,58);
- **Contrôle erroné**: Transmission d'une action différente de celle accomplie par l'utilisateur (e.g. l'utilisateur clique sur le bouton 1 et l'événement MouseMove est transmis à l'application);
- **Contrôle autonome**: un événement est transmis alors que l'utilisateur n'a pas accompli d'action sur le système (e.g. un événement MouseClick est transmis sans aucune intervention de l'utilisateur).

Description AADL des architectures

Les architectures présentées dans ce papier seront décrites dans cette section en utilisant la représentation graphique du langage AADL (Architecture Analysis and Design Language) [45]. AADL est un langage de description (textuel et graphique) standardisé en 2004 par la société SAE (Society of Automotive Engineers) pour aider à l'analyse et à la conception d'architectures système. Le langage AADL a principalement été utilisé pour décrire les systèmes temps réels complexes dans les systèmes avioniques et spatiaux mais il a aussi été utilisé

dans d'autres domaines. Nous l'utilisons dans cet article pour décrire graphiquement les architectures tolérantes aux fautes que nous proposons pour en évaluer la fiabilité. La description est adaptée pour le domaine d'application du cockpit par l'ajout d'une représentation du pilote, de ses interactions avec le système ainsi qu'un symbole «engrenage» pour représenter le traitement des données. La Figure 3 introduit les composants et les connexions en AADL utilisés dans ce papier.

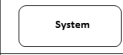

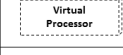
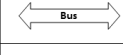
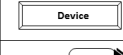

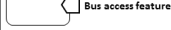
AADL component representation	Meaning of the component
	System : Hierarchical organization of components.
	Processor : Hardware component including a CPU, memory, bus ... and allowing software execution.
	Virtual processor : Logical resource allowing software execution.
	Bus : Provides physical connectivity between execution platform components.
	Device : Interface to external environment (typically physical components interfacing with the environment).
	Event Data Ports : Ports allowing directional transfer of queued messages.
	Bus access feature : To model required access to shared bus.

Figure 3. Composants AADL utilisés dans l'article

Une architecture tolérante aux fautes pour les composants interactifs

Dans cette section, nous présentons tout d'abord en détail une architecture auto-testable (mécanisme présenté rapidement au-dessus) qui permet d'assurer la **détection** des fautes, puis nous présentons deux architectures permettant leur **recouvrement** : i) une architecture N-auto-testable, où le recouvrement est assuré de manière autonome par le système et ii) une architecture avec recouvrement impliquant l'opérateur.

Une architecture auto-testable pour la détection de fautes appliquée aux composants interactifs

La Figure 4 présente l'architecture d'un composant usuel (Figure 4-a) comparé à un composant auto-testable (Figure 4-b). Cette figure rend explicite les différences entre les deux architectures. Elle est utilisée ici pour présenter la manière d'implanter les composants auto-testables pour des applications interactives. Pour cela, il faut suivre deux étapes : implantation du MON et son confinement par rapport au COM.

- Pour chaque composant interactif, on doit définir un MON et de nombreuses solutions sont disponibles. [47] a proposé d'appliquer un mécanisme auto-testable aux widgets en utilisant la seconde implantation présentée dans les travaux connexes. Il propose aussi l'utilisation d'une spécification formelle [33] pour décrire l'ensemble des composants du widget auto-testable (Dispatcher, COM, MON and Comparateur). Cette solution pose des problèmes d'implantation tels que la

spécification d'un composant qui autorise le dispatching des événements d'entrée. [20] a proposé d'appliquer la philosophie auto-testable à un niveau plus élevé, en utilisant la première implantation présenté dans les travaux connexes. Le challenge est ici la définition du programme (MON) vérifiant l'exécution du COM. Cette solution permet de grouper, par exemple, l'ensemble des moniteurs en un seul moniteur pour l'intégralité de l'application.

- Un mécanisme auto-testable ne suffit pas pour assurer la détection des fautes, il est donc nécessaire de confiner le COM du MON. En effet, une faute impactant un composant (e.g. le COM), peut également affecter le second composant (e.g. le MON). Cela serait le cas si tous les composants d'une architecture étaient exécutés dans la même partition. Le standard ARINC 653 [2] définit un partitionnement qui assure la ségrégation spatiale et temporelle pour le domaine de l'aviation civile. L'architecture proposée dans le papier rend explicite la ségrégation entre le COM et le MON, représentés ici par deux processeurs virtuels différents que l'on peut considérer comme deux partitions ARINC 653 séparées.

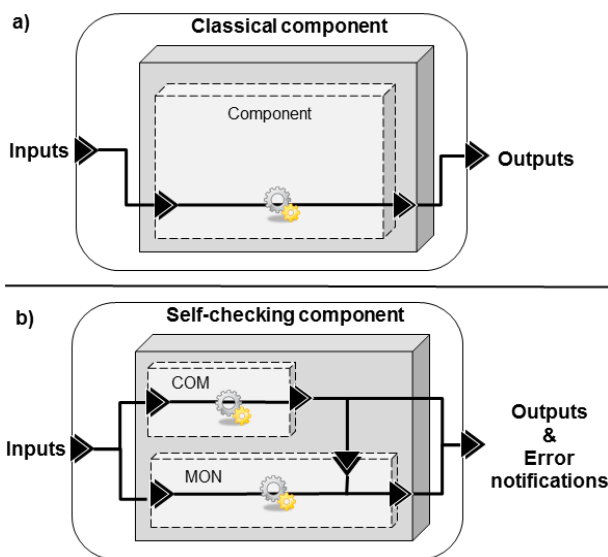


Figure 4. Architecture d'un composant usuel (a) comparée à celle d'un composant tolérant aux fautes (détection seule) (b)

Architectures pour le recouvrement de fautes appliquée aux composants interactifs

Les composants auto-testables sont capables de détecter la présence d'une faute et de déclencher la notification associée. Pour pouvoir ajouter du recouvrement et tolérer ainsi les fautes, plusieurs solutions de conception sont envisageables. Dans cet article nous en proposons deux, qui seront illustrées dans l'étude de cas :

- *L'architecture N-auto-testable (automatique)*
L'architecture N-auto-testable présentée dans la Figure 5 est composée de deux ou plus composants

auto-testables numérotés séquentiellement. Les événements d'entrée sont transmis à tous les composants auto-testables et sont traités en parallèle. Toutes les sorties et notifications d'erreur sont transmises à un mécanisme de décision. Si le premier composant fournit un résultat sans erreur, ce dernier est transmis tel quel en tant que résultat du composant N-auto-testable. Sinon, le mécanisme de décision ne prendra plus en considération le résultat du premier composant auto-testable et prendra à la place les résultats du suivant. Ceci se répète jusqu'à ce que le dernier composant défaille, ce n'est qu'à ce moment qu'une notification d'erreur est envoyée. Si tous les widgets auto-testables sont défaillants, le widget N-auto-testable ne fait plus que de la détection de faute, sans faire de recouvrement. Si un des widgets auto-testables traite une donnée sans erreur, la faute précédente est alors détectée, recouverte, et ne sera pas notifiée.

Cette architecture est particulièrement efficace car elle permet la diversification des composants auto-testables, ce qui pourrait se faire en les faisant développer par différentes équipes avec différents langages de programmation. Cette possibilité de diversification rend cette architecture très précieuse même si elle en devient naturellement plus chère à réaliser. Un autre aspect important repose sur le fait que le recouvrement est automatique et ne nécessite pas de support extérieur.

- *L'architecture impliquant l'utilisateur*
Il est possible de faire du recouvrement d'une autre manière en introduisant un mécanisme spécifique de gestion des notifications d'erreur par l'utilisateur. Dans ce cas, le principe est de notifier l'opérateur lorsqu'une faute est détectée par le système, laissant le recouvrement à la charge de l'opérateur. Cette approche présente l'avantage de garder l'opérateur dans la boucle de contrôle et lui permet donc de garder une marge de décision pour la résolution des fautes. Cependant, la charge de travail supplémentaire rend problématique cette approche lors de l'apparition de multiples fautes dans un temps limité. On voit ainsi apparaître un problème de conflit entre utilisabilité et fiabilité. Il est possible d'évaluer l'impact en termes d'utilisabilité (et plus précisément en termes d'efficacité) en décrivant et en analysant la complexité des tâches liées au recouvrement. Une approche systématique pour évaluer ce type d'impact a été testée sur 4 architectures de tolérance aux fautes différentes et peut être trouvée dans [22].

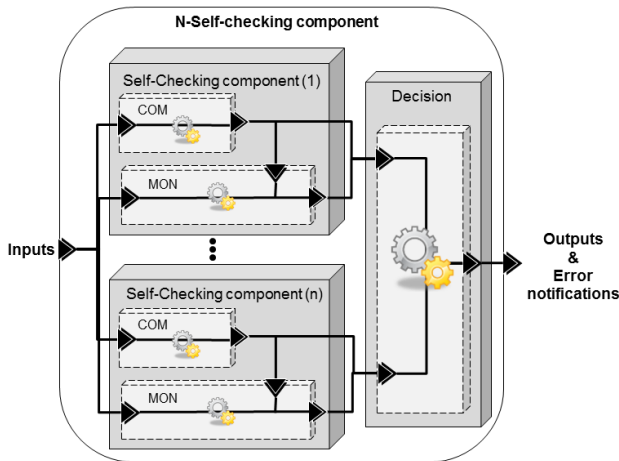


Figure 5. Une architecture tolérante aux fautes (incluant la détection et le recouvrement) d'un composant interactif

Une architecture pour des systèmes interactifs tolérants aux fautes

Lorsque l'on considère la fiabilité des systèmes informatiques, la fiabilité de l'ensemble du système est celle de l'élément le moins fiable. Nous avons proposé précédemment des architectures pour les composants interactifs capables d'effectuer la détection et le recouvrement des fautes. Si l'on suit l'architecture ARCH de la Figure 2, de tels mécanismes doivent être définis pour tous les composants i.e. les périphériques d'entrée et de sortie, les drivers, le gestionnaire de fenêtres, les widgets et l'application, c.à.d. le dialogue et le noyau fonctionnel.

L'ajout de mécanismes de tolérance aux fautes à ces composants soulève des problèmes du fait des contraintes temporelles imposées par la (courte) boucle de feedback immédiat nécessaire au gestionnaire d'événement des périphériques d'entrée. Nous travaillons sur ces aspects, à la fois dans le contexte des interactions WIMP et des interactions directes et multi-touch [23].

Comportement détaillé d'éléments interactifs tolérants aux fautes

Les travaux connexes proposent habituellement des

solutions génériques ayant un haut niveau d'abstraction comme celles que nous avons proposées dans la section précédente. Il faut bien sûr raffiner ces solutions lors des phases d'implantation pour permettre de décrire en détail les connexions entre les périphériques d'entrée-sortie et chacun des composants, ainsi que les connexions entre ces composants eux-mêmes. Il faut ensuite produire une description bas niveau de la variété des composants de l'architecture. De tels comportements peuvent être relativement complexes et leur intégration bien plus encore. [47] a démontré que l'utilisation de descriptions formelles dédiées aux systèmes interactifs, comme ICO [33] permet de décrire le comportement des widgets auto-testables. Une des difficultés est alors la production d'une seconde version du comportement pour respecter la diversification dans le but d'éviter les fautes de mode commun entre le COM et le MON comme détaillé dans [48].

APPLICATION DE L'ARCHITECTURE DE TOLERANCE AUX FAUTES A UNE ETUDE DE CAS

Pour illustrer l'approche proposée, nous présentons une mise en œuvre de l'architecture auto-testable sur une application interactive de cockpit appelée FCU Backup (Flight Control Unit Backup).

Description informelle

L'application FCU Backup est conçue pour permettre à l'équipage technique (pilote, copilote) d'interagir avec l'autopilote (AP) et de configurer les affichages de vol et de navigation. Cette application est affichée sur deux des afficheurs LCD du cockpit (un pour chaque pilote). L'équipage peut ainsi interagir avec l'application en utilisant un dispositif appelé Keyboard and Cursor Control Unit (KCCU en bas et à gauche de la Figure 6) qui combine un trackball et un clavier dans un unique composant matériel.

Architecture tolérante aux fautes pour le FCU Backup

La Figure 6 présente une vue détaillée de l'architecture (suivant l'architecture ARCH – cf. Figure 2) du FCU Backup, où le périphérique d'entrée est le KCCU et le périphérique de sortie est l'écran LCD du cockpit. Le gestionnaire de fenêtres permet de les connecter aux

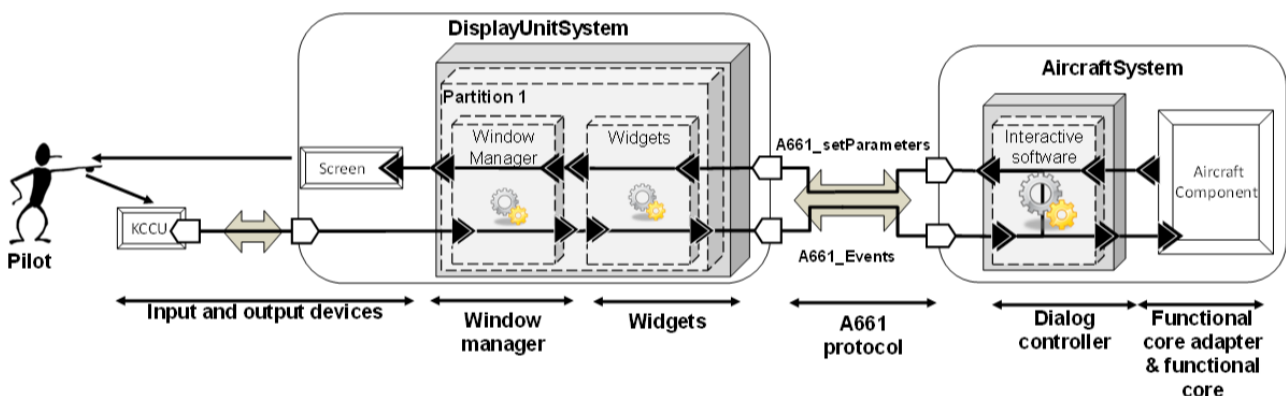


Figure 6. Architecture non tolérante aux fautes de l'étude de cas

widgets qui interagissent avec le contrôleur de dialogue via le protocole défini par le standard ARINC 661 [3]. Le contrôleur de dialogue est, quant à lui, directement relié aux systèmes avioniques.

Il est important de souligner que l'architecture doit respecter le standard ARINC 661 [3] puisqu'il est le seul standard utilisé dans le domaine des avions commerciaux civils. Il définit les widgets et leurs paramètres mais aussi les protocoles de communication entre les widgets et l'application à laquelle ils appartiennent. Les noms utilisés pour décrire les événements et leurs paramètres sont A661_Events et A661_setParameters comme décrit dans le standard. Dans de précédents travaux [5], chaque composant était modélisé en utilisant le formalisme ICO (excepté pour le gestionnaire de fenêtres pour lequel certaines parties ont été décrites en code JAVA) dans le but de traiter la fiabilité d'une telle application.

La Figure 7 présente l'extension de cette architecture pour la rendre tolérante aux fautes. Cette extension suit précisément l'architecture auto-testable présentée précédemment. Les widgets et le gestionnaire de fenêtres sont ici implantés comme des composants auto-testables auquel on a adjoint un composant MON. Ils sont donc capables d'envoyer des notifications d'erreur au contrôleur de dialogue qui contient des mécanismes de recouvrement. Le contrôleur de dialogue est lui aussi un composant auto-testable composé d'un COM et d'un MON. Du fait des contraintes d'espace, nous ne traiterons pas ici le contenu du composant moniteur. Il peut être implanté comme un moniteur unique ou encore comme une multitude de moniteurs assemblés, par exemple un pour chaque widget et un pour le gestionnaire de fenêtres. Sur notre prototype, nous avons testé les deux implantations présentées dans la section précédente (soit le mécanisme auto-testable automatique et celui mettant en jeu l'utilisateur). Plus d'information sur cette approche peut être trouvée dans [21].

Avec l'architecture mettant en jeu l'utilisateur, lorsqu'une faute est détectée, le système affiche un message pour que l'équipage fasse un redémarrage de l'application. Dans ce cas, si la faute détectée est transitoire (et donc non permanente) elle sera supprimée avec le redémarrage du système.

Avec l'architecture N-auto-testable, les composants logiciels de l'avion intègrent la gestion des erreurs déclenchées par les composants auto-testables tels que les widgets ou le gestionnaire de fenêtres. La Figure 7 représente ce comportement par les liens étiquetés *Error notifications* entre les boîtes des widgets et du contrôleur de dialogue ainsi qu'entre le composant de traitement d'erreur du système avionique situé dans la partition 3 et le MON du contrôleur de dialogue situé dans la partition 2. Les autres composants de l'application interactive (comme les périphériques d'entrée-sortie, le noyau fonctionnel etc.) doivent également être adjoints des mécanismes de tolérance aux fautes pour être résilients face aux fautes naturelles. Ce n'est pas présenté dans ce cas d'utilisation, mais on peut le réaliser en leur appliquant les architectures présentées dans ce papier (architecture auto-testable et N-auto-testable).

POSITIONNEMENT DE L'APPROCHE PAR RAPPORT AUX CONTRIBUTIONS ANTERIEURES

La Figure 7 rend explicite la séparation des composants dans les différentes partitions. Cette séparation est fondamentale pour assurer la tolérance aux fautes si l'on veut pouvoir prendre en compte les fautes communes entre le COM et le MON. La manière d'allouer les composants dans les partitions suivant le standard ARINC 653 [2] a été présentée dans [20]. Cette architecture est générique et peut être exploitée pour la conception de tout système interactif si la tolérance aux fautes fait partie des exigences. Toutefois, le comportement des composants peut-être aussi une source d'erreurs. Dans des travaux antérieurs, nous avons

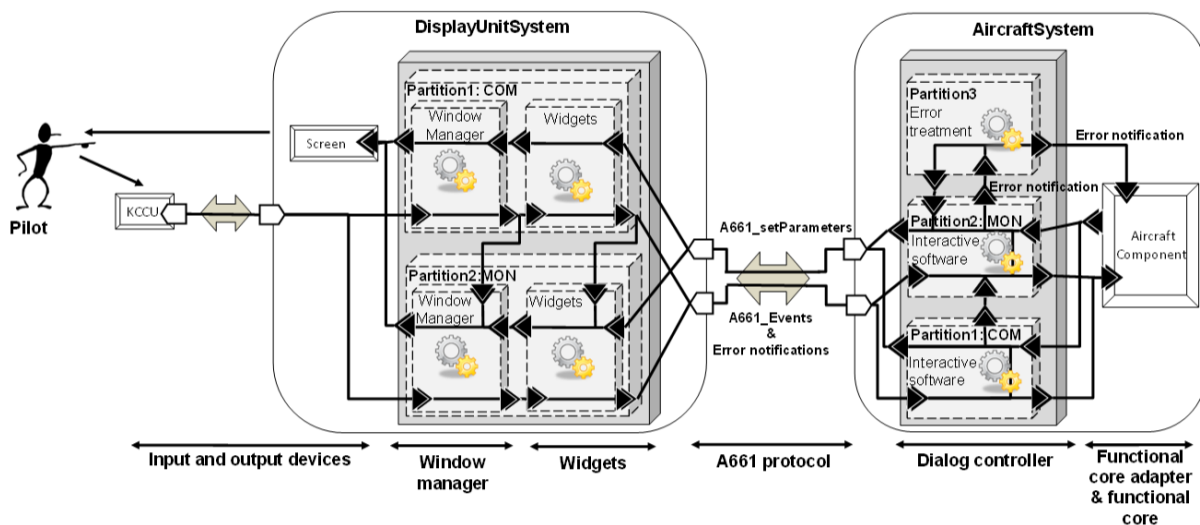


Figure 7. Architecture tolérante aux fautes appliquée à l'étude de cas FCU Backup

proposé une technique de description pour le comportement des widgets [5], des widgets auto-testables [47] ainsi qu'une description dissociée du COM et du MON (cette dernière étant à base de propriétés) [21]. Il est intéressant de noter que l'architecture proposée est l'aboutissement de travaux de longue haleine assurant la détection, le recouvrement et le confinement des fautes.

CONCLUSION ET PERSPECTIVES

Dans cet article nous avons proposé une taxonomie qui rassemble plusieurs travaux de recherche dans les domaines de la sécurité (comportements malveillants), de la tolérance aux fautes, des méthodes formelles ainsi que des facteurs humains qui sont 4 domaines traités généralement de manière séparée. Dans ce cadre, nous avons identifié les besoins pour résoudre les problèmes liés à la fiabilité des systèmes interactifs, et plus particulièrement, ceux se trouvant dans les systèmes critiques. Nous avons ainsi défini les contours d'une liste de problématiques à résoudre pour augmenter la fiabilité des systèmes interactifs.

La seconde partie de ce papier se concentrait sur une architecture auto-testable pour les systèmes interactifs, capable de détecter les fautes naturelles. Nous avons également proposé deux architectures candidates pour effectuer le recouvrement de telles fautes: la première est automatique tandis que la seconde est pilotée par un utilisateur (chacune ayant ses avantages et limitations détaillés dans l'article).

L'article a ensuite présenté comment de telles architectures peuvent être appliquées à un cas réel d'utilisation : l'application FCU Backup. L'application de telles architectures à ce cas d'utilisation permet alors la détection et le recouvrement des fautes matérielles pouvant affecter l'application. Cependant, l'adjonction de telles architectures soulève encore d'autres problèmes à identifier et à résoudre.

Premièrement, cet article s'est concentré sur les techniques d'interaction WIMP, l'application de telles architectures risque d'être plus problématique lors du passage aux interactions post-WIMP comme la manipulation directe ou les interactions multi-touch pour lesquelles les entrées et les sorties sont bien plus entrelacées.

Deuxièmement, l'ajout des composants nécessaires à la tolérance aux fautes (moniteurs, dispacheur, comparateurs...) aux applications interactives augmente significativement leur taille (approximativement par 5 pour les widgets auto-testables) même si l'ensemble des fonctionnalités reste constant. Il y a donc un besoin grandissant de développer des techniques de description formelle capable de passer à l'échelle et dédiées à la description des systèmes interactifs.

REMERCIEMENTS

Ces travaux ont été financés en partie par Airbus via le contrat R&T Display System X31WD1107313 et les projets FENICS, financés par CORAC.

REFERENCES

1. Accot J., Chatty S., Maury S. and Palanque P. Formal Transducers: Models of Devices and Building Bricks for Highly Interactive Systems 4th Eurographics workshop on "design, specification and verification of Interactive systems", 1997, Springer Verlag.
2. ARINC 653 Avionics Application Software Standard Interface. ARINC Specification 653. Airlines Electronic Eng. Committee July 15, 2003
3. ARINC 661 Cockpit Display System Interfaces to User Systems. ARINC Specification 661. Airlines Electronic Eng. Committee 2002.
4. Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C. Basic concepts and taxonomy of dependable and secure computing. In IEEE Trans. on Dependable and Secure Computing, vol.1, no.1, pp. 11- 33, 2004
5. Barboni, E., Conversy, S., Navarre, D., Palanque, P. Model-Based Engineering of Widgets, User Applications and Servers Compliant with ARINC 661 Specification. DSVIS 2006. LNCS 4323, pp. 25–38.
6. Basnyat, S, Palanque, P, Schupp, B, Wright, P (2007) Formal socio-technical barrier modelling for safety-critical interactive systems design (2007) Safety Science, Vol 45, n°5, June, ISSN: 0925-7535
7. Basnyat S., Chozos N. and Palanque P. Multidisciplinary perspective on accident investigation. Reliability Engineering & System Safety Volume 91, n° 12, 2006, Pages 1502-1520
8. Bass, L., Little, R., Pellegrino, R., Reed, S., Seacord, R., Sheppard, S., and Szezur, M. R. "The Arch Model: Seeheim Revisited." UI Developers' Workshop. 1.0 (1991).
9. Bastide, R., Palanque, P., Navarre, D., A Tool-Supported Design Framework for Safety Critical Interactive Systems, Interacting with computers, 2003, vol. 15/3, pp. 309–328.
10. Bastide, R., Sy, O., Palanque, P., Navarre, D. Formal specification of CORBA services: experience and lessons learned. ACM Conf. on Object-Oriented Prog. Sys. Lang. and Applications (OOPSLA'2000). ACM Press, p105-117.
11. Bass L., Clements P., Kazman R., Software Architecture in Practice, Addison Wesley, Reading, Mass., 1998.
12. Bowen J. and Reeves S.. 2011. UI-driven test-first development of interactive systems. In Proceedings of the 3rd ACM symposium on Engineering interactive computing systems (EICS '11). ACM, New York, NY, USA, 165-174.
13. Bowen J. and Reeves S. 2012. Modelling user manuals of modal medical devices and learning from the experience. ACM symp. on Engineering interactive computing systems (EICS '12). ACM, 121-130.
14. Bowen J. and Reeves S. Modelling safety properties of interactive medical systems. ACM symp. on Engineering interactive computing systems (EICS '13). ACM DL, 91-100.
15. Bowen J. and Stavridou V. Formal Methods, Safety-Critical Systems and Standards. Software Engineering Journal, 8(4):189–209, July 1993.
16. Campos, J and Harrison, M. D. Formally verifying interactive systems: A review. 4th EG workshop on Design, Specification and Verification of Interactive Systems DSVIS '97. 109-124. 1997. Springer Verlag.
17. Dearden, A. M and Harrison, M. D. Formalising human error resistance and human error tolerance. Proc. of 5th Int. Conf. on Human-Machine Interaction and Artificial Intelligence in Aerospace. 1995. EURISCO

18. Dessiatnikoff A., Nicomette V., Alata E., Deswarte Y., Leconte B., Combes A. and Simache C. Securing Integrated Modular Avionics computers, 32nd Digital Avionics System Conference (DASC), October 6-10 2013:
19. DO-254 - Design Assurance Guidance for Airborne Electronic Hardware, RTCA Inc, EUROCAE, April 2000.
20. Fayollas C., Fabre J-C., Navarre D., Palanque P. and Deleris Y. Fault-Tolerant Interactive Cockpits for Critical Applications: Overall Approach. 4th Int. Workshop on Software Engineering for Resilient Systems (SERENE 2012), LNCS, Springer Verlag. pp. 134-155.
21. Fayollas C., Fabre J.-C., Palanque P., Cronel M., Navarre D., Deleris Y. A software-implemented fault-tolerance approach for control and display systems in avionics. 20th IEEE Pacific Rim Int. Symp. on Dependable Computing. LNCS.
22. Fayollas C., Martine C., Palanque P., Deleris Y., Fabre J.-C., Navarre D. An approach for assessing the impact of dependability on usability: application to interactive cockpits. European Dependable Computing Conf.. pp 198-209. 2014
23. Hamon A., Palanque P., Silva J-L., Deleris Y. and Barboni E. Formal description of multi-touch interactions. Symposium on Engineering interactive computing systems (EICS '13). ACM, 207-216
24. Harrison M. & Dix A. A state model of direct manipulation. In M. Harrison and H. Thimbleby (eds.) Formal Methods in Human Computer Interaction, p. 129-151, Cambridge University Press, 1990.
25. Hecht H. and Fiorentino E. Reliability assessment of spacecraft electronics. In Annual Reliability and Maintainability Symp., pages 341-346. IEEE, 1987.
26. Hollnagel, E. Barriers and Accident Prevention. 2004. Ashgate.
27. IBM (1989) Common User Access: Advanced Interface Design Guide. IBM, SC26-4582-0
28. Ladry J-F, Navarre D., Palanque P. Formal Description Techniques to Support the Design, Construction and Evaluation of Fusion Engines for SURE (Safe Usable, Reliable and Evolvable) Multimodal Interfaces. Int. Conf. on Multimodal Interfaces (ICMI 2009), ACM, p. 135-142.
29. Laprie, J-C., Arlat, J., Béounes, C., Kanoun, K. Definition and Analysis of hardware and software Fault-Tolerant Architectures, IEEE computer, vol.23, no.7, pp.39-51, 1990.
30. Martinie C., Palanque P., Navarre D., Winckler M., and Poupart E. 2011. Model-based training: an approach supporting operability of critical interactive systems. Proc. of ACM symp. on Engineering interactive computing systems (EICS '11). ACM DL, 53-62
31. Masci, P., Ayoub, A., Curzon, P., Harrison M., Lee, I. and Thimbleby, H. Verification of interactive software for medical devices: PCA infusion pumps and FDA regulation as an example. ACM symp. on Engineering interactive computing systems (EICS '13). ACM DL, 81-90.
32. Navarre, D., Palanque, P., Basnyat, S., (2008) Usability Service Continuation through Reconfiguration of Input and Output Devices in Safety Critical Interactive Systems. 27th Int. Conf. on Computer Safety, Reliability and Security (SAFECOMP 2008) LNCS 5219, pp. 373-386.
33. Navarre, D., Palanque, P., Ladry, J., and Barboni, E. ICOs: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability, ACM TOCHI, 2009, V. 16, 4, pp. 1-56
34. Neema S., Bapty T., Shetty S. & Nordstrom S. 2004. Autonomic fault mitigation in embedded systems. Eng. Appl. Artif. Intell. 17, 7 (October 2004), 711-725.
35. Nicolescu B., Peronnard P., Velazco R., and Savaria Y. Efficiency of Transient Bit-Flips Detection by Software Means: A Complete Study. Proc. of the 18th IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT '03). IEEE Computer Society, 377-384.
36. Palanque P. & Basnyat S. Task Patterns for Taking into account in an efficient and systematic way both standard and erroneous user behaviours. 6th International Conference on Human Error, Safety and System Development, Springer Verlag pp. 123-139.
37. Palanque P., Barboni E., Martinie C., Navarre D., and Winckler M. A model-based approach for supporting engineering usability evaluation of interaction techniques. Proc. of ACM symp. on Engineering Interactive Computing systems (EICS '11). ACM DL, 21-30
38. Palanque P., Bastide R., Dourte L. Contextual Help for Free with Formal Dialogue Design. In proceedings of the Fifth International Conference on Human-Computer Interaction, (HCI International '93), 1993, Volume 2
39. Pnueli A Applications of Temporal Logic to the Specification and Verification of Reactive Systems: A Survey of Current Trends. LNCS n° 224 p.510-584. Springer Verlag 1986.
40. Polet, P, Vanderhaegen, F, and Wieringa, P. Theory of safety related violation of system barriers. Cognition Technology & Work, 4, 3, 171-179. 2002.
41. Rajkumar A. and Blandford A. 2012. A distributed cognition model for analysing interruption resumption during infusion administration. Proc. of 30th European Conf. on Cognitive Ergonomics (ECCE '12). ACM DL, 108-111.
42. Reason, J. (1990). Human Error, Cambridge University Press (et sa trad : L'erreur humaine, PUF Le travail humain, 1993)
43. Reiter M.K. and Stubblebine S. G. Authentication metric analysis and design. ACM Trans. Inf. Syst. Secur. 2, 2 (May 1999), 138-158.
44. RTCA-EUROCAE. Software Considerations in Airborne Systems and Equipment Certification. DO-178B / ED-12B, December (1992)
45. SAE-AS5506B: SAE Architecture Analysis and Design Language (AADL), Interanctional Society of Automotive Engineers, Warrendale, PA, USA (September 2012).
46. Schroeder B., E. Pinheiro, and W.-D. Weber. DRAM errors in the wild: a large-scale field study. In ACM SIGMETRICS, pages 193-204, Seattle, WA, June 2009.
47. Tankeu-Choitat, A., Navarre, D., Palanque, P., Deleris, Y., Fabre, J.-C., Fayollas, C. Self-checking components for dependable interactive cockpits using formal description techniques. In Proc of 17th IEEE Pacific Rim Int. Symp. on Dependable Computing (PRDC 2011), 10p.
48. Tankeu-Choitat A., FabreJ-C., Palanque P., Navarre D., Deleris Y. Self-Checking Components for Dependable Interactive Cockpits. Work. on Dependable Computing (EWDC 2011), ACM DL 10p.
49. Thimbleby H., Gimblett A. Dependable keyed data entry for interactive systems. Proceedings of the 4th Int. Workshop on Formal Methods for Interactive Systems (FMIS 2011)
50. Traverse, P., Lacaze, I. and Souyris, J. Airbus fly-by-wire: a total approach to dependability, Proc. IFIP World Computer Congress, pp. 191-212. 2004.
51. Wegner P. 1997. Why interaction is more powerful than algorithms. Commun. ACM 40, 5 (May 1997), 80-91.
52. Wright N., Patrick A. S., and Biddle R.. Do you see your password?: applying recognition to textual passwords. Proc. of Symp. on Usable Privacy and Security (SOUPS '12). ACM DL, 14 pages
53. Yau S.S, R.C Cheung, "Design of self-Checking Software", proc. Int. Conf. on Reliable Software, Los Angeles, CA, USA, IEEE Computer Society Press, 1975, pp. 450-457.
54. Yeh Y. C. (Bob), "Triple-Triple Redundant 777 Primary Flight Computer", IEEE Aerospace Applications Conf., pp 293-307, 1996

Conception d'un Dispositif pour Interagir avec des Données Multidimensionnelles : Disco

**Gary Perelman, Marcos Serrano,
Mathieu Raynal, Emmanuel Dubois**
Université de Toulouse - IRIT - Elipse
31062, Toulouse, France
<prénom.nom>@irit.fr

Célia Picard, Mustapha Derras
Berger-levrault
31676, Labège, France
<prénom.nom>@berger-levrault.fr



Figure 1. Disco est un dispositif à multiples degrés de liberté avec une base semi-sphérique inspirée du culbuto. Nous avons conçu différents facteurs de forme pour la moitié supérieure, de gauche à droite : Spherical, Rounded et Concave. Trois prises en main sont identifiées et étudiées dans l'article, de gauche à droite : Squeeze, Touch et Lay.

RESUME

En s'inspirant de travaux fondateurs proposant des souris à multiples degrés de liberté, cet article présente la conception d'un nouveau dispositif d'interaction basé sur le principe du culbuto : Disco. Nous présentons divers scénarii d'usage et explorons la manipulation de ce dispositif au travers de deux études. D'abord nous observons la prise en main de trois versions de Disco avec différents facteurs de forme. Ensuite nous étudions les capacités et limites liées à la translation, rotation (yaw) ou inclinaison (pitch, roll) physique de deux versions de Disco selon trois prises en main différentes. A partir des résultats nous proposons des guides de conception afin de créer des techniques d'interaction qui exploitent au mieux les différents degrés de liberté du dispositif pour interagir avec des données multidimensionnelles.

Mots Clés

Dispositif en entrée ; dispositif à multiples degrés de liberté ; multidimensionnel.

ACM Classification Keywords

H.5.2. Information interfaces and presentation: Interaction.

INTRODUCTION

Des travaux ont proposé des extensions de la souris soit pour intégrer des nouvelles modalités d'interaction [6,29] soit pour étendre ses degrés de liberté [5,14]. Les

premiers, comme l'intégration du multitouch [6] ou du touchscreen [29], ne modifient pas la manipulation physique de la souris, toujours limitée à des déplacements en 2D. Les seconds proposent d'étendre les degrés de manipulation de la souris en permettant les inclinaisons et les rotations, avec comme dispositifs fondateurs Rockin'Mouse [5] et VideoMouse [14]. Ces deux dispositifs ont montré leur intérêt pour interagir avec des environnements 3D ou des grands espaces d'information. Néanmoins ces dispositifs sont basés sur un facteur de forme proche de la souris qui ne favorise pas la rotation en raison du placement des boutons. De plus aucune étude n'a été menée sur les limites et les degrés de liberté de leur manipulation.

Inspirés par ces travaux fondateurs et leurs limitations, nous explorons dans cet article un dispositif original, appelé Disco, qui préserve les qualités de pointage de la souris dans les interfaces WIMP et intègre un plus grand nombre de degré de liberté que la souris. Disco est un dispositif en entrée basé sur le principe du culbuto (Figure 1) : la base du dispositif est hémisphérique et sa forme vue du dessus ronde. Il offre donc trois formes de manipulation : translation, rotation (yaw) et inclinaison (pitch et roll) (Figure 1). Nous explorons l'espace de conception de Disco puis étudions ses limites et capacités du point de vue des amplitudes possibles pour chaque manipulation et de la séparabilité des différentes manipulations.

Pour cela, nous explorons les principaux attributs de conception de ce dispositif. En prenant en compte des variations du facteur de forme de Disco basés sur le degré d'arrondi de la moitié supérieure du dispositif (concave, courbé ou hémisphérique - Figure 1), nous examinons dans une première étude les prises en main possibles pour ces différentes formes (Figure 1). Nous sélectionnons trois prises en main et deux formes

préférées par les utilisateurs. Dans une deuxième étude nous demandons aux participants de réaliser des translations, des inclinaisons et des rotations avec les prises en main et les formes sélectionnées. Les résultats nous permettent d'identifier les limites et les domaines de valeur des manipulations de Disco.

Nous réinvestissons ces résultats pour définir l'utilisation de Disco dans différents cas d'usage illustrant des applications à données multidimensionnelles [15,22] : navigation dans des menus, navigation sur une carte, paramétrage d'un outil de dessin et contrôle d'un jeu de simulation. Ces illustrations permettent de définir des mises en œuvre des Disco dans des contextes informatiques précis impliquant des dispositifs existants (souris, raccourcis claviers, SpaceMouse, etc.). Ils définissent autant de situations d'interaction devant faire l'objet d'évaluations expérimentales ultérieures.

Nos contributions sont : 1) l'exploration de l'espace de conception d'un nouveau dispositif sphérique, Disco, 2) l'étude des prises en main induites par trois variations du nouveau dispositif, 3) l'étude des limites et des valeurs des degrés de manipulation du dispositif et 4) l'application des résultats à la conception de techniques d'interactions pour plusieurs scénarii illustratifs.

ETAT DE L'ART

Nous situons d'abord Disco par rapport aux dispositifs en entrée multidimensionnels puis étudions les autres dispositifs de type souris à plusieurs degrés de liberté. Nous parcourons ensuite les travaux traitant des aspects physiologiques de la prise en main de dispositifs tangibles.

Dispositifs en entrée multidimensionnels

Différentes taxonomies permettent de classer les dispositifs en entrée : Foley [12] caractérise les dispositifs par les tâches qu'ils permettent de réaliser ; Buxton [9] propose de classer les dispositifs selon leurs propriétés physiques et les degrés de liberté ; Card [10] a généralisé la taxonomie de Buxton pour mieux décrire des dispositifs complexes. A partir de ces taxonomies, on peut classer les dispositifs selon deux types [11] : les dispositifs isotoniques basés sur le déplacement du dispositif pour modifier la position du pointeur (souris par exemple) ; et les dispositifs isométriques où l'on applique une force sur le dispositif pour déplacer le pointeur (souris3D par exemple) plus souvent couplés à un contrôle de vitesse. Toutefois ces derniers ne sont pas adaptés pour un pointage efficace dans des interfaces WIMP ce qui induit une interaction partagée entre la souris et le dispositif. Notre dispositif est à la frontière entre ces deux types car nous pouvons nous en servir à la fois comme dispositif isotonique (du fait des translations), mais aussi comme dispositif isométrique (grâce aux rotations et aux inclinaisons). Ce couplage est important pour avoir un dispositif universel qui soit utilisable quel que soit le contexte d'interaction (interface WIMP, environnement virtuel 3D, etc.) car il nous permet de conserver les avantages des dispositifs

isotoniques notamment pour le contrôle du pointeur dans un environnement 2D, mais aussi d'avoir tous les Degrés De Liberté (DDL) indispensables pour une interaction avec un EV3D.

Souris à plusieurs degrés de liberté

Augmenter les degrés de liberté de la souris, principalement à travers l'intégration de l'inclinaison ou de la rotation [5, 14], a été une solution initialement conçue pour faciliter l'interaction avec des environnements 3D. Rockin' Mouse [5] est un travail fondateur : il s'agit d'une souris avec des bords arrondis permettant d'incliner le dispositif en avant/arrière et sur les côtés, ajoutant ainsi deux degrés de liberté supplémentaires. Une version commerciale a été présentée en 2009, le Lexip 3DM. Ces deux dispositifs ne permettent pas la rotation (yaw) et leur usage n'a pas été étudié sur d'autres domaines que l'interaction dans des EV3D. VideoMouse [14] est un dispositif similaire à Rockin' Mouse, qui intègre en plus la rotation et a donc les 6 DDLs, En revanche, ses degrés d'inclinaison sont assez limités (+/-20°). Ces différents travaux étendaient la souris sans modifier sa forme (en dehors de l'arrondi des bords de la base) et aucune étude n'a porté sur le lien entre prise en main et liberté de mouvement du dispositif.

D'autres travaux se sont focalisé uniquement sur la rotation de la souris [2, 1]. Une solution pour détecter la rotation consiste à utiliser deux souris [2]. La rotation a été utilisée pour manipuler des objets 3D [2] et pour interagir avec des palettes d'outils contextuels [1]. Cependant, du fait que ces dispositifs ne proposent pas de gestion des inclinaisons, leurs DDLs sont limités à 4.

Facteurs humains sur la prise en main d'objets tangibles

Les travaux précédents n'ont pas étudié l'effet de la prise en main du dispositif sur ses degrés de liberté ou son usage. Par contre des travaux en psychologie [19] et physiologie [23] ont porté sur la préhension d'objets tangibles. Les travaux en psychologie montrent que le type de préhension (nombre et position des doigts) dépend autant de la forme de l'objet que de l'objectif de la tâche [19]. Par exemple nous ne prenons pas de la même façon une cuillère pour la tendre à quelqu'un que pour manger. Nous étudions donc dans notre travail différentes prises en main pour les trois actions principales de Disco : rotation, translation et inclinaison.

Afin de tirer parti des degrés de liberté de Disco, nous devons aussi prendre en compte les degrés de liberté physiologiques du poignet humain qui diffèrent selon les axes de rotation [23]. Notre deuxième étude vise précisément à identifier quelles sont les limites physiologiques de manipulation de Disco selon différentes prises en main.

Interaction sur et avec des dispositifs arrondis

De façon assez hétérogène, des travaux en IHM ont proposé diverses solutions d'interaction exploitant la

forme ronde [24, 27]. Parmi les dispositifs, nous pouvons citer PALLA [27], un dispositif sphérique conçu pour la manipulation 3D en l'air. Concernant l'interaction sur des surfaces rondes, Roudaut et al. [24] ont étudié l'interaction tactile sur des surfaces à différents degrés de courbure et sur différents points de la courbe. Les surfaces convexes (comme Disco) offrent une meilleure précision de pointage que les surfaces concaves. Ces résultats sont encourageants pour l'intégration future du toucher sur Disco.

Séparation des degrés de liberté

Habituellement, la manipulation d'objets virtuels sur différents degrés de liberté (translation, rotation) se fait de manière séparée et séquentielle [17]. C'est pourquoi, dans ce travail, nous étudions comment séparer les degrés de liberté de Disco pour pouvoir réaliser des actions indépendantes.

Néanmoins, ces actions sont indissociables dans le monde physique et il serait donc souhaitable de pouvoir les réaliser de façon combinée [28, 16]. Des travaux ont donc proposé des techniques d'interaction pour manipuler des objets virtuels qui combinent la rotation et la translation : Rotate'N Translate [17] est une technique conçue pour des tables tactiles basée sur l'usage d'un seul point de contact avec la surface interactive. Les études menées sur Rotate'N Translate montrent qu'elle est plus efficace que la manipulation traditionnelle en utilisant des actions séparées. Disco ayant 3 types de manipulations, il est envisageable de les combiner pour augmenter le nombre d'action ou optimiser leur accès.

ASPECTS DE CONCEPTION DE DISCO

Principe : le culbuto

Disco est un dispositif en entrée basé sur le principe du culbuto : sa base est hémisphérique (Figure 2-A) et sa forme vue de dessus est ronde (Figure 2-B). Cette demi-sphère est lestée de sorte que son centre de masse soit situé dans l'axe vertical du centre de la demi-sphère (Figure 2-C).

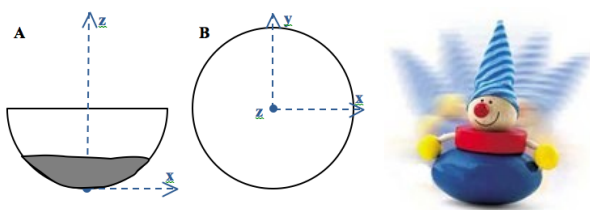


Figure 2. Base de Disco vue de face (a) et du dessus (b). Principe de recentrage du culbuto (c).

Les intérêts d'utiliser le principe du culbuto sont multiples. Tout d'abord sa forme permet d'offrir à l'utilisateur trois ensembles de manipulations de types différents : des translations, des inclinaisons (autour de l'axe X et Y, correspond au pitch et roll) et des rotations (autour de l'axe Z, correspond au yaw). Ainsi l'utilisateur dispose en un seul dispositif de nombreux modes d'interactions sans recourir à l'utilisation d'un dispositif supplémentaire ou à des combinaisons de

touches par exemple. De plus sa forme symétrique ne contraint nullement son utilisation selon une orientation prédéfinie, contrairement aux souris classiques. Enfin sa forme arrondie et le lest lui confèrent une capacité d'auto-recentrage, à l'image des dispositifs élastiques ou isométriques [30]. Du point de vue structurel, Disco dispose donc des caractéristiques d'un dispositif isotonique pour les rotations et les translations, et de celles d'un dispositif isométrique pour les inclinaisons (auto-centrage). Il combine ainsi les avantages et capacités d'une souris (amplitude, stabilité), d'une souris 3D (rotations) et d'un joystick (inclinaisons).

Ces principes constitutifs de DISCO amènent un certain nombre de constats et questionnements sur des aspects de la conception de Disco que nous développons ci-après.

Séparabilité des degrés de liberté

Tout d'abord, lors des inclinaisons de Disco, son centre subit inévitablement une translation dans la direction de l'inclinaison. De plus, la stabilité du dispositif est telle qu'elle garantit son auto-recentrage ; toutefois lors de sa manipulation en translation, par exemple, rien dans sa forme ne vient contraindre des oscillations (en inclinaison ou rotation) autour de sa position de référence. Comment dès lors distinguer une translation d'une inclinaison, et les inclinaisons (ou translations) voulues des inclinaisons (ou rotations) parasites ? Nous explorerons ces considérations dans l'expérience 2 que nous présentons dans la suite de cet article.

Forme et prise en main

La moitié inférieure du dispositif est basée sur le culbuto. Nous explorons différentes options de conception pour la moitié supérieure du dispositif. Cette forme doit permettre de poser la main dessus confortablement et doit être symétrique autour de l'axe Z (Figure 1) afin de conserver les propriétés de symétrie. Nous étudions trois formes basées sur des variations de l'arrondi supérieur (Figure 1): la moitié supérieure peut être concave (Disco Concave), légèrement convexe (Disco Rounded) ou hémisphérique (Disco Spherical). Notre première étude explore les prises en main possibles pour ces différents types de forme. Dans les trois cas, Disco est lesté de pâte à modeler.

Liberté des degrés de manipulation

En termes de manipulations physiques, du fait de la forme hémisphérique continue de Disco, les inclinaisons, rotations et translations peuvent être librement produites et combinées, quelle que soit la position de Disco. De plus par rapport au point de contact de référence de Disco avec la surface de travail, des inclinaisons sont possibles selon toutes les directions, de 0 à 360°, et leur amplitude peut varier de 0 à 90°. Par construction, l'angle de rotation est illimité. Dans l'usage, l'angle de rotation est limité selon le type de prise en main de l'utilisateur. Notre deuxième étude explore les limites de rotation du dispositif.

L'étude 1 explore les différentes prises en main envisagées par les utilisateurs pour les différentes manipulations permises par Disco.

ETUDE 1: PRISE EN MAIN

Etant donné la forme hémisphérique de Disco et la variation du degré d'arrondi envisagé dans la conception de celui-ci, la prise en main peut varier d'un utilisateur à l'autre. Le but de cette première étude est d'identifier les différents modes de prise en main que les utilisateurs adoptent lors d'une première confrontation aux trois variantes de Disco et en l'absence d'instructions spécifiques relatives à la prise en main de ce dispositif.

Tâche

La tâche confiée aux utilisateurs consiste à réaliser, avec la main dominante, des translations, inclinaisons et rotations libres (sans contraintes) de Disco, toujours en contact avec la table. La manipulation de Disco n'a aucun effet sur une application interactive et aucun feedback informatique n'est donc fourni aux participants.

Matériel

Le but étant d'étudier les types de prise en main des trois dispositifs, aucun matériel informatique n'est utilisé. Les participants manipulent donc les trois variantes en plastique et lestées de Disco présentées précédemment (voir Figure 1).

Procédure

L'étude suit un plan 4x3 avec Tâche (Translation, Inclinaison, Rotation ou Tout) et Forme (Spherical, Rounded et Concave) comme facteurs. L'étude se compose de deux étapes réalisées avec les 3 variantes de Disco (l'ordre des variantes est contrebalancé). Dans l'étape 1, le participant doit réaliser chacune des trois tâches (translation, inclinaison, rotation) pendant un temps limité et trouver la position qui lui semble optimale pour chaque manipulation. Dans l'étape 2, le participant doit choisir une seule prise en main lui permettant d'effectuer l'ensemble des trois tâches de manière optimale. Nous demandons au participant de réaliser tous les gestes possibles afin de valider sa prise en main. Enfin il est demandé à chaque participant de classer par ordre de préférence les trois formes étudiées. La durée moyenne de l'expérimentation était de 30 minutes.

Participants

Nous avons recruté 12 participants (2 femmes) âgés de 24,8 ans en moyenne (SD=1,8). 2 d'entre eux avaient de l'expérience dans l'utilisation de dispositifs possédant 4 degrés de libertés ou plus et un seul était gaucher.

Données collectées

Une photo de la main est prise pour chaque condition de prise en main. Au total, nous avons collecté 4 Tâches x 3 Formes x 12 Participants = 144 photos. Nous avons collecté l'ordre de préférence de chaque participant pour les trois formes étudiées.

Résultats

Préférence sur la forme de Disco

Nous analysons d'abord la préférence sur la forme du dispositif. Sur les 12 participants, 7 ont préféré Disco Rounded, 4 Disco Spherical et 1 seul le Disco concave. De plus Disco Concave a été classé dernier à 9 reprises. Les principaux inconvénients qui ont été relevés par les participants sont le peu de confort de cette forme du fait du trou au centre, et du manque d'accroches. Devant le faible succès de cette forme, nous avons choisi de ne pas la conserver pour la suite de nos études.

Disco Rounded est le plus apprécié car la base supérieure bombée, qui rappelle celle de la souris traditionnelle, permet de le prendre en main plus facilement et donne une impression de stabilité. Disco Spherical a lui l'avantage de permettre une position de main plus confortable. En revanche, la taille proposée pour Disco est, pour cette forme, un inconvénient car le dispositif Spherical est trop volumineux et la hauteur du dispositif a dérangé plusieurs participants.

Prises en main observées

Six types de prise en main ont été proposées par les participants (Figure 3).

Avec Disco Spherical, la prise en main 2 a été utilisée dans huit cas sur douze pour les inclinaisons et les translations. Pour les translations, aucune autre prise en main n'a été utilisée plusieurs fois. Pour les inclinaisons, la prise en main 3 a été utilisée à trois reprises. Pour les rotations, Disco Spherical a été pris en main de trois manières différentes : 1 a été utilisé dans la moitié des cas alors que 6 et 5 ont été chacun utilisés par 3 participants. Lors de la deuxième étape de l'étude, où les participants devaient adopter une position de main unique pour les trois manipulations, les prises en main 1, 2 et 3 ont été privilégiées avec respectivement quatre, quatre et trois participants qui les ont utilisées.

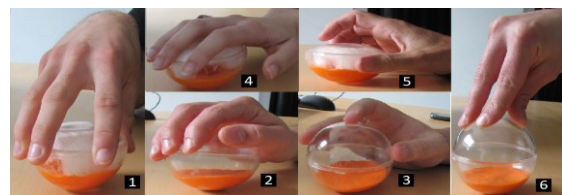


Figure 3. Les six types de prises en main utilisées par les participants sur différentes formes de Disco.

Concernant Disco Rounded, la prise en main 2 reste la plus utilisée pour les translations (huit fois sur douze). Pour les inclinaisons, 2 est utilisée moins souvent qu'elle ne l'était avec Disco Spherical (quatre fois pour Disco Rounded contre huit pour Disco Spherical), et les participants lui ont même préféré la prise en main 4 utilisée à cinq reprises. Enfin, pour les rotations, la prise en main 1 a été très fortement utilisée (dix cas sur douze). Malgré une forte utilisation des prises en main 2 et 1 lors de l'étape 2, ces préférences ne se sont pas retrouvées aussi marquées lors de l'étape 3 avec Disco

Rounded. En effet, même si 1 et 2 ont été les plus utilisés avec respectivement quatre et trois participants, l'écart est faible avec d'autres types de prise en main comme 3 et 4 utilisés chacun deux fois.

Synthèse

Il ressort donc de cette première étude que quelle que soit la forme de Disco, les prises en main 2 et 1 sont souvent privilégiées : nous les avons appelées « Lay » et « Squeeze » respectivement. « Lay » est caractérisée par le placement de la paume de la main sur Disco ; « Squeeze » correspond au placement des doigts sur le contour de Disco tout en ayant la paume relevée. Une troisième prise en main « Touch » a été conservée également. Elle est caractérisée par la présence de trois doigts sur le dessus de Disco tandis que deux autres sont positionnés de chaque côté de Disco, potentiellement au contact avec la table et/ou maintenant Disco (cf. Figure 1).. Cette prise en main correspond à des situations observées à de nombreuses reprises dans lesquels plusieurs doigts étaient posés sur Disco mais pas la main complète.

ETUDE 2: MANIPULABILITE DU DISPOSITIF

Le but de cette seconde étude est de cerner les capacités, limites et conditions d'utilisation de Disco. Concrètement l'objectif est d'identifier les amplitudes moyennes, minimales et maximales des différentes manipulations ainsi que les conditions dans lesquelles ces manipulations peuvent être utilisées distinctement, i.e. indépendamment les unes des autres. Ces résultats constitueront des enseignements clairs pour guider l'implémentation de techniques d'interaction utilisant Disco.

Tâches et instructions

Nous utilisons les trois prises en main choisies dans l'étude précédente (Figure 1). Le participant doit réaliser pour chaque prise en main des translations dans huit directions (nord, nord-est, est...), des inclinaisons dans les huit mêmes directions et les deux rotations (gauche, droite). Les directions sont inscrites sur la table d'expérimentation (Figure 4). Pour les translations, deux distances sont testées, représentés par deux cercles sur la table (Figure 4). Pour chaque translation, il est demandé à l'utilisateur de se déplacer, à partir du centre, dans une direction donnée jusqu'à dépasser le cercle indiqué puis de revenir au centre. L'instruction pour les inclinaisons et les rotations est de faire la manipulation de plus grande amplitude possible tout en demeurant confortable. Tout comme dans l'étude 1, il était rappelé de laisser Disco en permanence au contact de la table et la manipulation de Disco n'avait aucun effet sur une application interactive. Aucun feedback informatique n'était donc fourni aux participants : les tâches à réaliser étaient uniquement des manipulations physiques de Disco précisément énoncées par l'expérimentateur.

Matériel

Nous avons utilisé les deux formes de Disco préférées par les participants dans la première étude : Disco

Rounded et Disco Spherical. Pour localiser la position, orientation et inclinaison du Disco nous avons utilisé un système de vision OptiTrack avec 12 caméras infrarouges (précis à 0.01° et 1mm près). Nous avons rajouté sur Disco des billes réfléchissantes (placées de façon à ne pas gêner la prise en main du Disco ni son mouvement) afin de permettre aux caméras de détecter le dispositif (Figure 4).

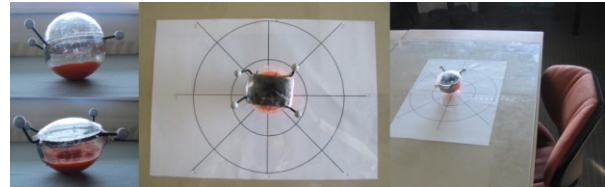


Figure 4. Apparatus utilisé dans l'étude. A gauche, les deux variantes de Disco avec les marqueurs infrarouges. A droite, le guide des directions de manipulation.

Participants

Nous avons recruté 12 participants (1 femme) âgés de 25,7 ans en moyenne (SD=3,8) et tous droitiers. 1 d'entre eux avait de l'expérience dans l'utilisation de dispositifs possédant 4 degrés de libertés ou plus. 10 d'entre eux avaient réalisé l'étude précédente.

Procédure

L'étude suit un plan 2x3x3 avec Forme (Rounded, Spherical), Prise en main (Squeeze, Lay et Doigt) et Manipulation (Translation, Rotation, Inclinaison) comme facteurs. Nous contrebalançons Forme et Prise en main pour former les 6 blocs de l'étude. Un bloc se compose de 8 translations courtes, 8 translations longues, 8 inclinaisons et 2 rotations dans un ordre aléatoire. Après chaque essai, Disco est remis en position initiale. Avant de commencer l'étude, l'utilisateur dispose d'une phase d'entraînement et de familiarisation où il réalise une fois chaque geste. Une longue familiarisation n'était pas nécessaire car les sujets impliqués dans cette étude ont tous été impliqués dans l'étude 1 ou des pré-tests. L'étude a duré 45 minutes en moyenne.

Mesures / données collectées

Nous avons enregistré les données complètes de 2 Formes x 3 Prises en main x (16 Translations + 8 Inclinaisons + 2 Rotations) x 12 sujets = 1872 gestes. A l'issue de chaque bloc, la fatigue physique associée au couple Forme/Prise en main était évaluée.

Résultats

Nous présentons d'abord les données collectées par manipulation (rotation, inclinaison et translation) puis la mesure de la fatigue. Les données obtenues n'ayant pas une distribution paramétrée, nous avons utilisé deux tests statistiques non paramétriques afin d'obtenir les résultats que nous vous présentons : le test de Wilcoxon et le test de Friedman.

Rotation

Un test de Wilcoxon a été utilisé pour déterminer si les différentes variantes de Disco ont un effet sur l'amplitude de la rotation à gauche ou à droite. Les résultats montrent

qu'il n'y a pas d'effet significatif entre les deux versions (Rotations droites ; $p=0.1866$; Rotations gauche : $p=0.3303$). Nous reportons donc les valeurs moyennes des deux variantes confondues.

La rotation moyenne observée, toutes directions confondues, est de $53,16^\circ$. La rotation moyenne vers la gauche est de $57,54^\circ$ alors que celle vers la droite est de $47,95^\circ$. Afin de déterminer si la direction de la rotation (gauche ou droite) à un effet sur l'amplitude de celle-ci, un test de Wilcoxon a été utilisé. Le résultat obtenu permet d'affirmer que les rotations vers la droite ont une amplitude plus faible que celles vers la gauche (Figure 5; $p=0,0017$).

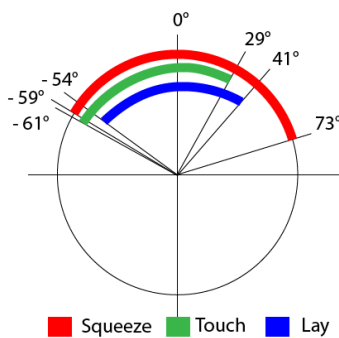


Figure 5. Amplitude des rotations par prise en main.

D'après les résultats de la première étude, la prise en main la plus souvent choisie pour les rotations est la position Squeeze. Les résultats obtenus via un test de Friedman viennent confirmer le fait que cette prise en main permet une plus grande amplitude de rotation que les deux autres (66.4° contre 45.15° et 47.97° ; $p=0,0017$). En revanche, il n'y a pas de différence significative entre les deux autres prises en main.

Inclinaison : amplitude

Pour déterminer si les variantes de Disco ont un effet sur l'amplitude maximale d'inclinaison, un test de Wilcoxon a été utilisé. Le résultat montre que Disco Spherical permet de plus grandes inclinaisons que Disco Rounded ($p \ll 0.001$).

En ce qui concerne Disco Spherical, la moyenne des inclinaisons maximales toutes prises en main confondues est de 40° pour toutes les directions sauf pour le Sud où elle est de 35° . Pour Disco Rounded, les directions: Nord-Est, Est et Sud-Est offrent 40° d'amplitude maximale moyenne toutes prises en main confondues alors que les directions: Nord, Sud et Nord-Ouest offrent 35° .

Pour les deux variantes de Disco évaluées, un test de Friedman a été effectué afin de déterminer si certaines prises en main avaient une amplitude d'inclinaison plus élevée (Figure 6).

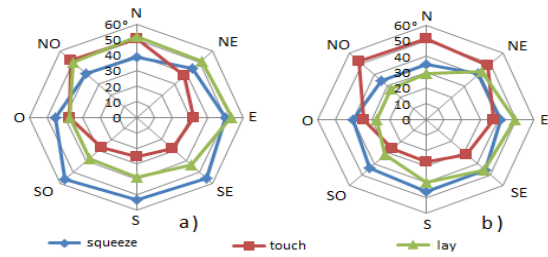


Figure 6. Amplitude des inclinaisons par direction pour a) Disco Spherical et b) Disco Rounded

Pour Disco Spherical (Figure 6-a), la prise en main Touch permet, en moyenne, des amplitudes d'inclinaison plus faibles que les deux autres prises en main ($p=0,0001$). Il est important de noter que pour le Disco Spherical, les amplitudes d'inclinaisons permises par la prise en main Touch sont quasiment toutes plus faibles que celles de la prise en main Lay. Cette dernière peut donc être utilisée à la place de Touch. En ce qui concerne le Disco Rounded, ces deux prises en main sont complémentaires. La distribution des inclinaisons par prise en main est similaire à celle du Disco Spherical, sauf pour la prise en main Lay. Dans ce cas précis, des inclinaisons d'amplitude 40° peuvent être réalisées dans les directions Nord, Nord-Est, Est, Sud-Est et Nord-Ouest et des inclinaisons d'amplitude 35° dans les trois dernières directions (Figure 6-a).

Pour Disco Rounded (Figure 6-b), les trois prises en main ne sont pas significativement différentes globalement. Pour la prise en main Touch, des inclinaisons d'amplitude 40° peuvent être réalisées dans les directions : Nord, Nord-Ouest et Nord-Est et des inclinaisons d'amplitude 35° dans la direction de l'Est. Pour la prise en main Lay, des inclinaisons d'amplitude 40° peuvent être réalisées dans les directions : Nord-Est, Est et Sud-Est et des inclinaisons d'amplitude 35° dans la direction du Sud. Pour la prise en main Squeeze, des inclinaisons d'amplitude 35° peuvent être réalisées dans les directions Nord et Nord-Ouest. Pour toutes les autres directions, cette prise en main permet des inclinaisons maximales de 40° (Figure 6-b).

Inclinaison : écart

Les résultats obtenus montrent un écart entre la direction d'inclinaison demandée et la direction d'inclinaison effectuée. Cet écart peut être plus grand d'un côté de la direction demandée que de l'autre. Ainsi, pour les directions Nord, Nord-Est, Est, Sud et Nord-Ouest, l'écart est plus important dans le sens des aiguilles d'une montre (de 10.19° à 16.63°). Pour les autres directions, c'est dans le sens inverse que l'écart est le plus important (de 7.60° à 11.29°). Il n'y a pas d'écart significatif entre les versions de Disco.

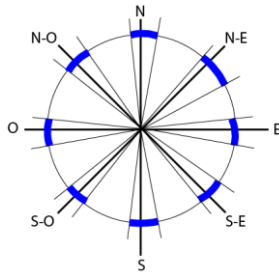


Figure 7. Ecart entre la direction d'inclinaison demandée et la direction effectuée.

Translation

Nous avons enregistré les données d'inclinaison et rotation involontaires de Disco au cours des translations.

L'amplitude maximale moyenne des inclinaisons involontaires est de 12.46°. Nous utilisons un test de Wilcoxon afin de déterminer si les versions de Disco ont un effet sur ces inclinaisons : Disco Spherical induit des inclinaisons involontaires significativement plus grandes que Disco Rounded Figure 8-a ; $p < 0.001$). Un test de Friedman montre que la prise en main n'a pas d'influence significative sur ces inclinaisons involontaires (Figure 8-b).

Deux translations étaient effectuées par les participants au cours de l'étude : courtes (14cm) et longues (28cm). Un test de Wilcoxon a permis de déterminer que les inclinaisons involontaires sont plus élevées lors d'une translation plus grande (Figure 8-c; $p < 0.001$). Cet effet est significatif pour chaque forme de Disco (translations courtes : $p=0,2264$; translations longues : $p=0,0717$).

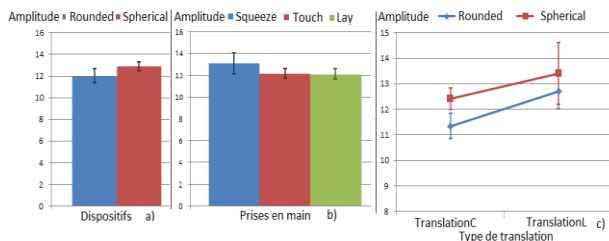


Figure 8. Amplitude des inclinaisons involontaires a) par Disco, b) par prise en main et c) par type de translation

L'amplitude moyenne des rotations gauche ou droite involontaires est de 14.93°. En utilisant la même démarche que pour les inclinaisons, un test de Wilcoxon détermine un effet des versions de Disco sur les rotations involontaires: Disco Spherical provoque des rotations involontaires avec une amplitude plus élevée que Disco Rounded (Figure 9-a ; $p=0,0099$). Un test de Friedman montre que la prise en main n'a pas d'influence significative sur les rotations involontaires (Figure 9-b).

Un test de Wilcoxon montre que les rotations involontaires sont plus élevées lors de la translation longue toutes prises en main et versions de Disco confondues ($p=6,107 \times 10^{-12}$). Cet effet reste vrai pour chaque variante de Disco (Figure 9-c ; $p=0,0408$).

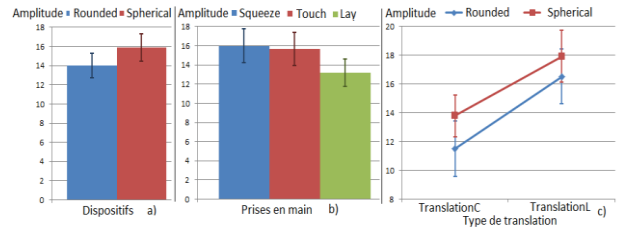


Figure 9. Amplitude des rotations involontaires a) par Disco, b) par prise en main et c) par type de translation

Fatigue

La fatigue a été calculée en utilisant une échelle de Borg [8]. La valeur moyenne est de 11.72. Un test de Wilcoxon montre que les variantes de Disco utilisées n'ont pas d'effet sur la fatigue physique ressentie ($p=0.9909$). Concernant les prises en main, un test de Friedman prouve que la prise en main Squeeze est plus fatigante que la prise en main Lay ($p=0.0144$).

Synthèse

Cette étude nous a permis d'obtenir les limites et les domaines de valeurs des trois degrés de manipulation des deux versions de Disco : rotation, inclinaison et translation.

L'amplitude de rotation est plus importante à gauche qu'à droite. De plus, la prise en main Squeeze permet une amplitude plus élevée.

La variante de Disco que l'on utilise a un effet significatif sur l'amplitude d'inclinaison. Les prises en main ont aussi un effet sur ces amplitudes : la prise en main Touch sur Disco Spherical est plus limitée que les autres prises en main. Les écarts observés entre la direction d'inclinaison demandée et celle effectuée varient selon les directions étudiées.

La longueur des translations a un effet sur les amplitudes d'inclinaison et de rotation : les translations longues induisent plus de mouvements involontaires. La variante de Disco utilisée a un effet sur les mouvements involontaires : Disco Spherical induit plus d'erreurs.

Finalement, la fatigue physique ressentie n'est pas dépendante de la variante de Disco utilisée. En revanche, la prise en main Squeeze est plus fatigante que Lay.

SCENARI D'UTILISATION

Du fait de ses caractéristiques principales, les usages possibles de Disco sont multiples. Nous illustrons ces possibilités en particulier dans le contexte de la navigation dans des menus et cartes, du dessin, d'exploration d'espaces 3D et de jeux en montrant comment les résultats de nos études influencent les techniques d'interaction couplées à DISCO.

Marking menu

Parmi les composantes de base d'une application interactive figurent les menus. De nombreuses formes de menus ont été proposées [3]. Utiliser Disco dans ce contexte fournit une option originale pour permettre la manipulation physique de ces différentes formes de

menus comme par exemple les marking menus [18]. En considérant un marking menu proposant huit items de premier niveau, chacun est activé par une inclinaison de Disco dans la direction correspondante. Compte tenu des résultats de l'étude, quelque soit la forme et la prise en main de Disco, il apparaît que ce mapping est parfaitement possible puisque les plages d'inclinaison mesurées autour de chaque direction n'excède pas 22,5 degrés de chaque côté de la direction visée. Par conséquent, huit items adjacents de 45 degrés chacun sont manipulables par Disco avec des inclinaisons.

Disco-Flower menu

Considérons maintenant les flower menus [4]. Dans chacune des huit directions d'un flower menu, sept courbes différentes peuvent être produites après la translation : avec une souris, ces deux mouvements (translation, courbe) doivent être produits de manière séquentielle. Disco-Flower menu est une variation des Flower menu qui offre la possibilité de produire les deux mouvements simultanément : les huit directions initiales sont réalisées par des translations de Disco et les sept profils de courbes sont contrôlés par le degré d'inclinaison de Disco pendant la translation. Les valeurs des inclinaisons pour chaque courbure doivent être fixées au-delà du seuil identifié dans la seconde étude, correspondant aux angles maximums atteints par des inclinaisons involontaires au cours des translations.

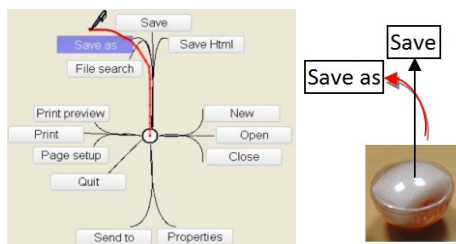


Figure 10. A gauche, Flower menu (tiré de [Bailly 08]). A droite, translation et inclinaison pour réaliser en un seul geste la commande "Save as" avec Disco-Flower menu.

Carte Google Earth

La navigation avec Disco peut aussi être étendue à la manipulation de données géographiques. Les rotations de Disco ajustent le niveau de zoom dans une carte, les inclinaisons permettent de déplacer librement le point de vue de la carte vers le Nord, l'Est, le Sud, Sud-Ouest, etc. et les translations permettent alors la sélection de points ou de zones dans la partie affichée [20, 13]. Pour conserver un contrôle isotonique sur le zoom comme dans Tilty Table [20], une prise en main maximisant les rotations (Squeeze) est utilisée. Concernant les inclinaisons, c'est également avec cette prise en main que les amplitudes maximales sont atteintes dans toutes les directions. Enfin la forme Spherical de Disco maximise significativement les inclinaisons. La prise en main Squeeze du Disco Spherical est donc à privilégier pour optimiser la navigation dans Google Earth.

Outil de dessin

Disco permet de manipuler les paramètres d'un outil de dessin au cours de son utilisation. Pour réduire les allers-retours entre le dessin et les outils, des lentilles et palettes d'outils ont été proposées [7], parfois dans des contextes bimanuels et tangibles [25]. Disco permet de lier chacune de ses manipulations à l'un des outils fréquemment utilisés. Les translations sont utilisées pour dessiner les tracés de crayon. Des inclinaisons selon les différentes directions permettent d'accéder rapidement à des fonctions de traitement du dessin en cours (remplissage, suppression, mode sélection/dessin...). Les rotations portent alors deux attributs offrant plusieurs valeurs possibles : les couleurs et le type de traits. Compte tenu du caractère asymétrique de l'amplitude des rotations accessibles, nous disposons les couleurs, plus nombreuses, à gauche où l'amplitude moyenne constatée est d'environ 60 degrés. Coté droit, les principaux types de traits sont accessibles directement (mode position) et les modes moins fréquemment utilisés sont accessibles en mode vitesse à partir d'un angle de rotation de 30°. Des contrôles de la position puis de la vitesse sont donc combinés comme suggéré dans [26].

Jeu de simulation

Enfin, considérons un simulateur, dans lequel, en fonction de l'élément sélectionné, des options de configuration sont offertes. C'est notamment le cas dans des systèmes de monitoring, mais aussi dans des jeux de simulation de vie. Prenons par exemple la configuration de son avatar dans le jeu Sims : coiffure, haut, pantalon et chaussures peuvent être modifiés pour changer le look de son personnage (Figure 11).

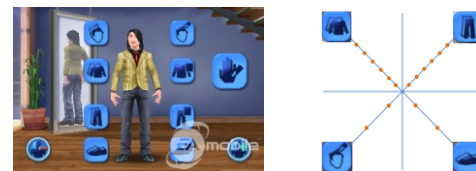


Figure 11. A gauche, menu par défaut dans le jeu Sims. A droite, menu en étoile optimisé pour Disco.

Pour chacun d'eux, Disco associe une direction d'inclinaison: le degré d'inclinaison permet de parcourir les possibilités. Considérant que le nombre de coiffures et de chaussures disponibles est nettement inférieur au nombre de hauts et de pantalons, l'étude menée suggère de placer les hauts et les pantalons sur les axes N, N-E, O ou N-O d'un Disco Spherical car ce sont les axes permettant les plus fortes amplitudes. A l'inverse les options plus limitées en nombre et concernant les chaussures et la coiffure pourront être placées sur les axes E, S-E ou S. Un design exploitant ces recommandations peut alors en plus prôner une symétrie des axes utilisés ainsi qu'une répartition homogène entre attributs du haut du corps et attributs du bas du corps comme illustré dans la Figure 11.

DISCUSSION

Contrôle du mouvement : position et vitesse

Il est possible d'utiliser un contrôle de la position ou de la vitesse pour gérer les translations et les rotations. Pour les inclinaisons, s'agissant d'un dispositif isotonique, un contrôle de la vitesse est privilégié [31] mais compte tenu de l'amplitude des inclinaisons possibles, et contrairement à une souris 3D par exemple, une utilisation pour un contrôle des positions est possible. Disco peut donc être manipulé en translation, inclinaison et rotation pour contrôler des positions (position control) ou des vitesses (rate control).

Disco sur la main dominante : étendre la souris

Notre étude porte sur la prise en main de Disco avec la main dominante pour augmenter la souris. Afin de réaliser les fonctions de la souris, Disco doit permettre la translation et la validation (boutons sur la souris).

Translation

Disco permettant différentes manipulations, trois questions se posent sur l'utilisation des translations pour contrôler un pointeur de souris : 1) est-ce que la translation de Disco déclenche des rotations ou inclinaisons accidentelles et 2) est-ce que une rotation ou 3) une inclinaison de Disco à l'arrêt déclenche une translation accidentelle. Cela revient donc à déterminer les conditions de séparabilité des trois manipulations.

Nous avons étudié la translation et établi les valeurs des inclinaisons et rotations accidentelles. Cela nous permet de définir des seuils afin de réaliser la translation sans déclencher des activations d'autres fonctions, liées à la rotation ou à l'inclinaison, par inadvertance.

Concernant les translations fortuites lors d'une rotation de Disco, l'étude ne nous a pas permis de les mettre en évidence. Par contre, l'inclinaison de Disco provoque, par construction, une translation du centre de masse du dispositif sur le plan de la table. Pour compenser les translations induites par les inclinaisons, une solution consiste donc à soustraire de la translation mesurée, la longueur de l'arc entre le point de contact initial de Disco avec la surface et le point final.

Validation

La validation est essentielle pour reproduire la fonction de la souris: l'ajout d'une surface tactile de petite taille de type touchpad, ou encore d'un bouton sur le volume hémisphérique est donc requis. Dans ce dernier cas une façon de conserver la symétrie de Disco consiste à ajouter un bouton circulaire sur la partie supérieure, similaire à la surface interactive ronde des iPod première génération (Figure 12). Une autre solution est de placer un bouton autour du diamètre de Disco, similaire au bracelet interactif WatchIt [21] (Figure 12). On pourrait utiliser une surface capacitive accompagnée de ressorts physiques situés sous la surface de Disco. Il serait ainsi possible d'obtenir un retour haptique des actions effectuées sur la surface de Disco tout en ajoutant des boutons de type « Apple Magic Mouse ».

Implémentation de Disco

Nous avons implémenté une première version fonctionnelle de Disco en utilisant un capteur magnétique sans fil (Polhemus – Patriot) (Figure 12). Ce capteur, placé à l'intérieur de notre prototype, permet de récupérer la position et l'orientation 3D du dispositif. Nous avons utilisé ce prototype pour implémenter une application de dessin et la sélection de menus linéaires.



Figure 12. Illustration de l'emplacement possible de boutons circulaires (droite) et prototype fonctionnel intégrant un capteur magnétique 3D et un bouton simple.

CONCLUSION ET PERSPECTIVES

Nous présentons dans cet article la conception d'un nouveau dispositif en entrée basé sur le principe du culbuto et nommé Disco. Il offre trois manipulations (translation, rotation et inclinaison) à multiples degrés de liberté. Nous avons tout d'abord observé les différentes prises en main possibles pour trois variantes du dispositif. Pour chacune des prises en main nous avons ensuite étudié les capacités limites en amplitude et séparabilité sur deux versions de Disco. Les résultats montrent que Disco permet des rotations asymétriques avec d'avantage d'amplitude sur la gauche ; Disco permet des inclinaisons variables selon les axes d'inclinaison et les prises en main, de 30° jusqu'à 57° en moyenne ; enfin la translation de Disco implique des inclinaisons et rotations accidentelles que nous mesurons afin de proposer un seuil pour éviter les faux positifs. Nous avons enfin réinvesti ces résultats pour définir l'utilisation de Disco dans différents cas d'usage : la sélection de commande, la navigation dans une carte, le dessin et un environnement de simulation.

Ce travail représente une première étape dans la conception de techniques d'interaction utilisant Disco. Nous allons d'abord nous focaliser sur la construction d'un prototype fonctionnel sans fil avec un bouton circulaire pour la validation/sélection. Nous comparerons ensuite Disco à différents dispositifs (SpaceMouse, Souris, etc.) afin d'évaluer ses performances dans des contextes d'utilisation précis tels que le pointage 2D et la manipulation d'objets 3D. Nous allons par la suite valider expérimentalement l'usage des techniques proposées notamment dans le contexte des scénarii d'utilisation évoqués. Enfin, nous allons travailler sur des extensions de Disco, par exemple l'intégration d'un feedback visuel (un mini-écran sur la partie supérieure).

REMERCIEMENTS

Ces travaux ont été réalisés dans le cadre du contrat de collaboration de recherche IRIT-Berger Levrault (N° CNRS 099856). Les auteurs remercient Gilles Bailly, Laura Jourdain, Christina Fernandez et les membres de l'équipe ELIPSE pour leurs suggestions.

BIBLIOGRAPHIE

1. Almeida, R. 2008. Palettes satellitaires: faciliter les changements d'outils avec la rotation de la souris. In Proc. of IHM '08. ACM, 173-176.
2. Almeida, R. and Cubaud, P. 2006. Supporting 3D window manipulation with a yawing mouse. In Proc. of NordiCHI '06. ACM, 477-480.
3. Bailly, G., Lecolinet, E. and Nigay, L. 2009. MenUA: A Design Space of Menu Techniques. www.gillesbailly.fr/menua/
4. Bailly, G., Lecolinet, E. and Nigay, L. 2008. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In Proc. of AVI '08. ACM, 15-22.
5. Balakrishnan, R., Baudel, T., Kurtenbach, G., and Fitzmaurice, G. 1997. The Rockin'Mouse: integral 3D manipulation on a plane. In Proc. of CHI '97. ACM, 311-318.
6. Benko, H., Izadi, S., Wilson, A., Cao, X., Rosenfeld, D., and Hinckley, K. 2010. Design and evaluation of interaction models for multi-touch mice. In Proc. of GI '10. CIPS, 253-260.
7. Bier, E.A., Stone, M.C., Pier, K., Buxton, W. and DeRose, T.D. 1993. Toolglass and magic lenses: the see-through interface. In Proc. of SIGGRAPH '93. ACM, 73-80.
8. Borg, G. Borg's Perceived Exertion and Pain Scales. Human Kinetics (1998), viii 104pp.
9. Buxton, W. Lexical and pragmatic considerations of input structures. Comput. Graph., 17(1) :31-37, 1983.
10. Card, S., Mackinlay, J., and Robertson, G.. 1990. The design space of input devices. In Proc. of CHI '90. ACM, 117-124.
11. Casiez, G. (2004). Contribution à l'étude des interfaces haptiques. Le DigiHaptic: un périphérique haptique de bureau à degrés de libertés séparés. Ph.D. Thesis, Université Lille 1, October 2004.
12. Foley, J., Wallace, V., and Chan, P. The human factors of computer graphics interaction techniques. IEEE Comput. Graph. Appl., 4(11) :13-48, 1984.
13. Harrison, S., Minneman, S., Back, M., Balsamo, A., Chow, M., Gold, R., Gorbet, M., Mac Donald, D. 2001. Design: the what of XFR: eXperiments in the future of reading. *interactions* 8, 3 (May 2001), 21-30.
14. Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R., and Conway, M. 1999. The VideoMouse: a camera-based multi-degree-of-freedom input device. In Proc. of UIST '99. ACM, 103-112.
15. Hurter, C., Conversy, S., Vinot, J-L., and Jestin, Y. 2009. Représentations écologiques de données temporelles: exemples et apports. In Proc. of IHM '09. ACM, 33-42.
16. Jacob, R., Sibert, L., McFarlane, D. and Preston Mullen, M. Integrality and separability of input devices. ACM TOCHI, 1, 1 (1994), 3-26.
17. Kruger, R., Carpendale, S., Scott, S., and Tang, A. 2005. Fluid integration of rotation and translation. In Proc. of CHI '05. ACM, 601-610.
18. Kurtenbach, G. and Buxton, W. 1993. The limits of expert performance using hierarchic marking menus. In Proc. of CHI '93 ACM, 482-487.
19. Marteniuk, R. G., MacKenzie, C., Jeannerod, M., Athenes, S., and Dugas, C. Constraints on human arm movement trajectories. Canadian Journal of Psychology 41, 3 (1987), 365--378.
20. Minneman 2001 Last accessed on YouTube: May 2014, <http://www.onomy.com/blue/tilty.html>, 2001
21. Perrault, S., Lecolinet, E., Eagan, J., and Guiard, Y. 2013. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. In Proc. of CHI '13. ACM, 1451-1460.
22. Perin, C., Christie, M., Vernier, F., and Lino, C. 2013. CollaStar: Interaction collaborative avec des données multidimensionnelles et temporelles. In Proc. of IHM '13. ACM, 115-125.
23. Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. 2009. Tilt techniques: investigating the dexterity of wrist-based input. In Proc. of CHI '09. ACM, 1943-1952.
24. Roudaut, A., Pohl, H., and Baudisch, P. 2011. Touch input on curved surfaces. In Proc. of CHI '11. ACM, 1011-1020.
25. Spindler, M., and Dachsel, R. 2009. PaperLens: advanced magic lens interaction above the tabletop. In Proc. of ITS '09. ACM, Article 7, 1 pages.
26. Tsandilas, T., Dubois, E., Raynal, M. 2013. Modeless Pointing with Low-Precision Wrist Movements. In Proc. of INTERACT 2013. Springer, p. 494-511.
27. Varesano, F. and Venero, F. 2012. Introducing PALLA, a novel input device for leisure activities: a case study on a tangible video game for seniors. In Proc. of FnG '12. ACM, 35-44.
28. Wang, Y., MacKenzie, C., Summers, V. and Booth, K. 1998. The structure of object transportation and orientation in human-computer interaction. In Proc. of CHI '98 . ACM, 312-319.
29. Yang, X-D., Mak, E., McCallum, D., Irani, P., Cao, X., and Izadi, S. 2010. LensMouse: augmenting the mouse with an interactive touch display. In Proc. of CHI '10. ACM, 2431-2440.
30. Zhai, S. 1998. User performance in relation to 3D input device design. SIGGRAPH Comput. Graph. 32, 4, 50-54.
31. Zhai, S., Human Performance in Six Degree of Freedom Input Control, 1995, University of Toronto

Adaptive Hand-Tracked System for 3D Authoring

Alexis Heloir
LAMIH-UMR CNRS 8201,
Valenciennes, France
DFKI-MMCI, Saarbrücken
alexis.heloir@univ-
valenciennes.fr

Fabrizio Nunnari
DFKI-MMCI, Saarbrücken,
Germany
fabrizio.nunnari@dfki.de

Christophe Kolski
LAMIH-UMR CNRS 8201,
Valenciennes, France
christophe.kolski@univ-
valenciennes.fr

ABSTRACT

We present the interaction design and the component architecture of an adaptive authoring system based on a consumer-range 3D input device. We claim that this system can help both novice and experienced users performing authoring tasks in a 3D authoring environment. The system uses a keyboardless self-adaptive interaction controller built upon a rule-based system that learns and infers the user's behavior/condition on the fly according to her actions; rearranging rules when necessary and suggesting breaks to avoid performance drops caused by fatigue or the so-called gorilla-arm effect.

Key Words

Gestural input, 3D authoring

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Animated characters are paramount in interactive media, where movement quality and gesture repertoire has now become an intrinsic characteristic of the essence of a character, at the same level as appearance, story background and skills. Animated characters might be a privileged support of the narrative and they also have the capability to support the expression of deaf individuals on the internet, by allowing them to express opinions anonymously in their primary language: Sign Language. All these applications – customization of character's movement, animated narrative and sign language generation – share a common requirement: they all need a convenient and straightforward 3D animation authoring system.

Recent advances in consumer-range interaction devices like the Kinect¹ or the Leap Motion² has opened the door to an unprecedented range of new user interfaces, interaction modalities and metaphors where gesture and bodily interaction are the cornerstones [10]. Taking inspiration

1. <http://www.xbox.com/kinect> (25 Feb 2014)

2. <http://www.leapmotion.com> (25 Feb 2014)

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670456>

from these trends, we propose a self adaptive architecture that has the potential to assist users in 3D authoring task. We currently focus on position objects in 3D space along 6 degrees of freedom. We believe that the interaction metaphors we propose in this paper can help user perform 3D editing in the following applications: character customization, animated narrative, and sign language synthesis.

Our major contribution consists of a keyboardless self-adaptive interaction controller built upon a rule-based system that learns and infers the user's behavior/condition on the fly according to her actions; rearranging rules when necessary and suggesting breaks to avoid performance drops caused by fatigue or the so-called gorilla-arm effect.

RELATED WORK

Animation is a highly cross-disciplinary domain which spans across acting, psychology, movie-making, computer graphics, and programming. Therefore, learning the art of traditional animation requires time, effort, and dedication. However, recent consumer-range technology has proved to be capable of enabling inexperienced users to either author animation of human-like bodies or to interactively control physical and digital puppets. [9, 7]

When an animation is realized using a specialized input device or motion capture system, is often referred to as performance-driven animation or puppeteering [9]. In performance-driven animation, the animator interactively operates a dedicated interface that allows him or her to control a synthetic character with a reinterpretation of the animator's original performance. For instance, Sanna et al. [7] used the Microsoft Kinect as an input-device for driving the locomotion of an animated character.

Other work proved that specific input devices and interactive user interfaces can improve the authoring and animation process. On the one hand, some devices are better suited to record straight-ahead animation by capturing the motion or the dynamics of the user [2], on the other hand, other interaction devices are more suited to locate and rotate objects in a 3D space [5]. Only a few systems are actually suited to both methods [4] and to our knowledge, no proposed architecture accounts for the possibility to switch seamlessly between the two modes during an editing session. We believe that the system we present has the potential to fit both modes during an edit session. Even if, in the current paper, we focus on object displacement, the hand tracking device we currently use could also be used to record the velocity profile of the user's hand.

The level of maturity of 3D UI design principles is still

lacking behind those for 2D graphical user interfaces. as stated in Bowman et al in 2013 [1]: “There is currently no standard 3D UI (and it’s not clear that there could be, given the diversity of input devices, displays, and interaction techniques), and few well-established guidelines for 3D UI design. While general HCI principles such as Nielsen’s heuristics [6] still apply, they are not sufficient for understanding how to design a usable 3D UI.” For many novice users, effective operation of 3D UIs requires going through a learning phase, even if these UIs are well designed. Even if it might seem unrealistic to completely get rid of such a learning phase, we would like to test the relevance of a self adaptive architecture for 3D UI. This architecture would be, at the beginning, very error-tolerant and easy to interact with, yet rather inefficient in terms of efficiency and accuracy. The UI would however accompany the user from apprentice to mastery, reconfiguring it’s interaction mechanics and thresholds, achieving better efficiency as the user gains confidence and proficiency while interacting with the UI. Previous work on adaptive UIs indeed shown that increasing accuracy led to strongly improved satisfaction. Increasing accuracy also resulted in improved performance and higher utilization of the adaptive interface. [3]

INTERACTION DESIGN

With a mouse-based interface or a multitouch screen, users can control at most three to four degrees of freedom at the same time ([X, Y, scroll] or [X, Y, pinch, rotate]). In contrast, a 3D input device such as the Leap Motion gives simultaneous access to 6 degrees of freedom: translation and rotation along the three axis. We thus expect users to be faster when directly manipulating the 3D scene rather than when using the mouse and the keyboard. Indeed, for single target selection, Sears and Shneiderman [8] have shown that direct-touch outperforms the mouse.

We design our prototype using the Blender³ 3D editing tool. The standard editing mechanism of Blender uses a combination of mouse and key strokes to select the mapping between the 2 DOFs of the mouse and the spatial properties of a virtual object. The editing system is albeit quite complex, and works as follows⁴. By pressing the key G (Grab) the object location in space and the mouse position are stored. From this moment the editor enters a modal state where the 2D offset of the mouse on the real desktop is mapped to a 3D translation of the virtual object. The translation occurs on a virtual plane parallel to the screen projection plane. During the interaction, the use of keys X, Y and Z can constraint the movement of the object to a different reference axes (x,y,z) or planes (xy, xz, yz). Hitting the left mouse button terminates the modal control, sticking the object to its current location. Hitting the right mouse button, or the ESC key, cancel the modal control, bringing the object back to its initial location. This last case does not appear in the figure for clarity reasons. A similar behavior is achieved by pressing R (rotate), where the default rotation is around an axis perpendicular to the projection plane, and x,y,z constraints work

3. <http://www.blender.org>, 6 Aug 2014

4. A comprehensive description of the editing basics can be found in the early tutorials of the Blender software.

similarly to the grab operation.

The interaction mechanism we implemented for the Leap Motion is much simpler: After selecting a virtual object, the user presses T to begin a simultaneous modal control for both rotation and translation, i.e 6 DOFs. We call the phase within the begin of the modal control and its confirmation/cancellation an *editing action*. During early trials, we observed a number of phenomena suggesting that performance editing with the HT&M approach can further improve together with the reliability of tracking technologies. For instance, when performing docking tasks, the user has a tendency to grab the virtual object as if he/she was holding a physical object.

These observations suggest that we should design an interaction mode based solely on the hand tracking device, aiming at increasing the performance and eliminating the need to learn keyboard commands. Finally, we believe that self-adaptation can be useful among different users as well as during long working sessions of a single user. In the latter case, the self-adaptation will tune the interaction mechanic of the system according to the fatigue condition of the user. The following section describes the design we propose to endow self-adaptation capabilities to our system.

SELF-ADAPTIVE ARCHITECTURE

The overall architecture of our interactive system is summarized in Figure 1. It follows a feedback-controlled loop model where the user input (hand motion) is filtered out and analyzed by a component called the Motion Analyzer. This component infers a set of mid-level motion primitives and sends them to the interaction manager: a reconfigurable rule based system in charge of triggering the right interaction mode according to its input motion primitives. The interaction manager delivers a flow of edit actions. This flow is continuously analyzed by the Status and performance assessor.

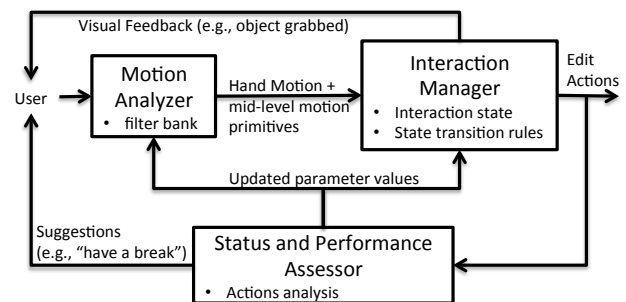


Figure 1. Our architecture is inspired by a feedback controller pattern

Our system should adapt to the user, whether the user is experienced or novice. Furthermore, if the user is novice, the system should accompany the user from beginner to master level. Doing so, it should enforce best-practices and monitor user’s progress while adapting to the user’s performance improvement. The system should also be capable of recognizing performance level deterioration and take measures against it. Again, by enforcing good practices or by suggesting pauses. Indeed gestural interaction, when used extensively, might induce fatigue, sometimes

called “gorilla arm”⁵. To avoid this, the assessor maintains a vector of descriptors that could be viewed as a simplistic user model accounting for proficiency and fatigue level. When the fatigue level increases too much, the system suggests a short break to the user. Also, the Status and Performance Assessor is continuously tuning the rules of the interaction manager to improve the user’s comfort and level of performance.

Motion Analyzer

The activity of the Motion Analyzer is based on the information stream received from the hand tracking device. Each data frame contains, among others, information about the number of detected hands and a set of data-structure instances storing the position and the orientation of each palm and each detected fingertip.

The Motion Analyzer filters out and analyzes the flow of frames streamed from the Leap. It consists of a set of primitive functions performing the analysis on a time-sliding window buffer of the Leap frames received in the last 2 seconds. These functions are predicates and a selection of them can be arranged according to the instructions delivered by the Status and Performance Assessor. These predicates indicate if the user’s hand appeared in the current frame (*newHand()*), of if the hand disappeared (*handLost()*). Some predicates have parameters, for instance the function telling if the hand has been stable for a certain amount of time supports two parameters: time span and distance threshold (*isHandStable[lookback_time, stability_radius]()*). Furthermore, *handFastMovement(threshold_speed)* detects if the user made a fast movement, which should be interpreted as moving away the hand from the editing task.

Interaction Manager

The Interaction Manager is an online reconfigurable rule-based/production system in charge of switching to the right interaction state according to the values computed by the mid-level motion primitives introduced in the previous section. The system handles three interaction states, as depicted in the right side of Fig. 1:

- HOVER: this state is active when a hand has been detected by the tracking system, but no editing action is actually carried on,
- GRAB: active when the user has selected an object and is moving it (visual hint),
- IDLE: active when no hand is visible by the tracking device.

When defined, new motion analysis primitives are bound to a selection of dynamic variables. They mostly represent time and distance thresholds. The value of these variables is updated on the fly according to how the user performs.

A selection of rules govern the basic state transition triggered by the presence/absence of the hand. When a new hand is detected, the system switches to state HOVER, regardless of its previous state. Similarly, when a hand tracking is lost, the system switches to the IDLE state.

5. http://en.wikipedia.org/wiki/Touchscreen#.22Gorilla_arm.22 (20 Jan 2014)

The rest of the interaction is based on the principle that when the user wants to start an editing action he needs to stabilize her hand into the sensor action space. When the hand is stable for enough time, the selected 3D object will start following the hand. This rule requires the user hand to be somehow far from the position where an object was dropped (GRAB state exited) in order to avoid undesired re-grabbing. This is done with a call to the *isDistantFromLastDrop()* primitive and was inserted according to preliminary observation of users’ behaviour. Stabilizing the hand again terminates the editing action.

Self-adaptivity – Status and Performance Assessor

In the previous section, we saw how the state-transition was governed by a set of rules in the Interaction Manager. These rules can be modified by reparameterizing the primitive functions composing the rules and their arrangement. Such changes have an influence on the interaction dynamics. The goal of the Status and Performance Assessor is to guarantee that the current rule arrangement and parameterization maximize the user’s comfort and efficiency.

We monitor aspects of the user interaction, such as her ability to keep her hand still, to control the velocity profile of the movement, as well as the reaction time to visual cues. This architecture is user-agnostic, which means that it does not build and track a model of the user, it rather tracks the short-term evolution of the user interaction and apply adaptation strategies in order to either increase the user’s efficiency or to limit the decrease of performance level.

We now present how the Status and Performance Assessor monitors the user’s performance level. Table 1 lists the variables we use to measure user performances. The values of the variables are calculated through the observation of the last $N_o = 25$ recognized actions. In the following, we describe each assessment variable and its influence on the Interaction Manager’s ruleset parameterization.

Here, we apply the principle behind closed feedback loops systems, where the output (user performance) is constantly monitored the amplification (or attenuation) of the input signal (user actions) until, with a certain delay and oscillation, the system becomes stable. So, as the user performance is observed to increase, the thresholds will be decreased, to accommodate user need of system reactivity. On the contrary, if the user performance is decreasing, thresholds will be relaxed to make the system less prone to timeouts and movement range exceeding.

The value e is the exponential moving average of the last 25 edit actions. If Y_i is the duration of the last edit action, e_i is computed as follows:

Table 1. List of variables used to assess user performance

Name	meaning
e	action edit time
m	action linear hand movement
c	cancelled actions
l	lost hand tracking
f	fast hand movement detected

$$e_i = \alpha \cdot Y_i + (1 - \alpha) \cdot e_{i-1}$$

The coefficient α represents the degree of weighting decrease, a constant smoothing factor between 0 and 1. A higher α discounts older observations faster. Since α may be expressed in terms of N time periods, where $\alpha = 2/(N + 1)$. And we want N to be around 25, we use a value of 0.077 for α .

The evolution of the resulting e value gives us a hint about the user's ability to perform faster or slower edit actions. If e decreases, we assume that the user needs a more responsive system. Therefore, we decrease the value of the GRAB_START/STOP_STABILITY_TIME parameters (Rules 6 and 9). On the opposite, if e increases, we increase the value of the GRAB_START/STOP_STABILITY_TIME parameters. Practically, e is used as an adjustment for a feedback gain which multiplies GRAB_START/STOP_STABILITY_TIME by $(1 + e_i)$ at each iteration.

m is computed exactly the same way as e on the cumulative distance which is traveled by the hand during an action. Since an experienced user is capable of accomplishing an edit with only a few large actions (GRABS) of the hand in space, a decreasing value of m indicates the user tendency to perform longer movements, including large positioning (hand is fast) and fine positioning (hand is slow) of the manipulated object. This suggests that the user is gaining in efficiency and that the interactions rules may be accommodated by decreasing the GRAB_START/STOP_STABILITY_RADIUS required to enter and exit the GRAB status. In other words GRAB_START/STOP_STABILITY_RADIUS are multiplied at each iteration by the following feedback gain: $(1 + m_i)$.

The value of c depends on how many of the last N_o operations have been canceled. An operation is canceled when the user presses the ESC key in order to restore the 3D object in its initial position. For each operation with $1 < i < N_o$, we consider $c_i = 0$ if the operation has not been canceled and $c_i = 1$ if it did. We calculate the tendency by interpolating a line among the sampled results. The tendency c' is calculated as the tangent of the interpolated line. We consider a positive tendency as the fact that too many operations started when the user didn't really mean to do. This will be used to increase both the GRAB_START/STOP_STABILITY_TIME and RADIUS thresholds.

The value of l is calculated, similarly to c , by counting how many times the hand tracking has been lost while performing the last N_o operations. A positive tendency tells us that the user hand is exiting too often from the editing space. This means that the user hardly feel uncomfortable in extreme hand positions. We use this as hint that we can increase the sensibility of the overall system, i.e., increase the ration between the quantity of motion performed by the dragged 3D object with respect to the same quantity of motion performed by the hand in real world. Consequently, also the value of DROP_OFF_DISTANCE (rule

6) and DIRECTION_CHANGE_SIZE (rule 8) are modulated.

The value of f is calculated similarly to c and l , by counting how many times the GRAB state has exited because a fast hand movement has been detected. If this occurs too frequently, the system modulates the value of the FAST_MOVEMENT_SPEED variable (rule 7), and may suggest the user to have a short break so that she could recover from the fatigue that might be induced by the gorilla ram effect. We are conducting further tests involving long edit session to correctly adjust this variable.

CONCLUSION AND PERSPECTIVES

To sum up, we have presented an animation system that has the potential, on the one hand, to enable novice users to author complex 3D edits and/or animations of humanoid characters and on the other hand, to increase the productivity of experienced users. This system is built upon both the Leap Motion and provides an intuitive authoring interface that has the potential to support two different edit schemes: performance capture and pose-to-pose animation. We further proposed a self-adaptive architecture inspired by a closed-loop controller that monitors the user's performance and tune on the fly the interaction controller in order to either increase the user's efficiency or to limit the decrease of performance. Since the device we used is commercially available at a consumer-range price and since we used open source software to build this system, we are publishing on-line⁶ all the sources that are necessary to build and reproduce the described architecture. We will assess the benefit brought by such a system in further work.

BIBLIOGRAPHIE

1. Bowman D. A. *The Encyclopedia of Human-Computer Interaction: 3D User Interfaces, 2nd Ed.* Interaction Design Foundation, 2013.
2. Chai J. & Jessica Hodgins J. K. Performance animation from low-dimensional control signals. In *Proc. of ACM Transactions on Graphics (SIGGRAPH 2005)* (2005).
3. Gajos K. Z., Everitt K., Tan D. S., Czerwinski M. & Weld D. S. Predictability and accuracy in adaptive user interfaces. In *In CHI'08*, ACM Press (2008).
4. Kipp M. & Nguyen Q. Multitouch puppetry: creating coordinated 3D motion for an articulated arm. ACM Press (2010), 147.
5. Lin J., Igarashi T., Mitani J., Liao M. & He Y. A sketching interface for sitting pose design in the virtual environment. *IEEE Transactions on Visualization and Computer Graphics* 18, 11 (2012), 1979–1991.
6. Nielsen J. & Molich R. Heuristic evaluation of user interfaces. In *Proc. CHI '90*, ACM (1990), 249–256.
7. Sanna A., Lamberti F., Paravati G. & Rocha F. D. A kinect-based interface to animate virtual characters. *Journal on Multimodal User Interfaces* (2012).
8. Sears A. & Shneiderman B. High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies* 34, 4 (1991), 593–613.
9. Sturman D. Computer puppetry. *IEEE Computer Graphics and Applications* 18, 1 (1998), 38–45.
10. Wigdor D. *Brave NUI world: designing natural user interfaces for touch and gesture*. Morgan Kaufmann, Burlington, Mass, 2011.

6. <http://slsi.dfki.de>

Un espace de caractérisation pour l'édition de courbes à trois dimensions

Thibaut Jacob¹, Gilles Bailly¹, Eric Lecolinet¹, Raphael Foulon², Etienne Corteel²

¹Télécom Paristech, CNRS LTCI
46, rue Barrault
75013 Paris
{prenom.nom}@telecom-paristech.fr

²Sonic Emotion
42 bis, rue de Lourmel
75015 Paris
{prenom.nom}@sonicemotion.com

RÉSUMÉ

La création et l'édition de courbes 3D intervient dans de nombreux domaines d'application comme la CAO, l'édition de contenus spatio-temporels dans une vidéo ou encore la génération de terrains et de routes. Cette diversité a conduit à une dispersion des travaux au sein de plusieurs communautés telles que SIGCHI ou SIGGRAPH. Cet article propose un espace de caractérisation visant à rassembler les techniques proposées pour la création et l'édition de courbes 3D. Cet espace est décomposé en deux axes: *dispositif* et *langage* afin de décrire et comparer les techniques existantes.

Mots Clés

Techniques d'interaction ; courbe 3D ; reconstruction géométrique ; création de trajectoire ; navigation 3D

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous.

INTRODUCTION

La création de courbes 3D est un sous-domaine de l'interaction homme-machine visant à la construction et à l'édition de courbes dans un environnement à trois dimensions. Créer une courbe 3D est une tâche courante avec la plupart des logiciels de CAO [Adobe Illustrator, AutoCAD], lesquels permettent généralement de créer des formes arbitraires sans passer par les primitives prédéfinies (cube, sphère). C'est également le cas des outils d'animation [Adobe Edge Animate] et de modélisation 3D, qui nécessitent de définir la trajectoire de la caméra [Blender] ou d'un objet dans une scène 3D. Enfin, certains systèmes de navigation 3D ou même de création musicale (édition de son 3D [41, 46, 26]) reposent aussi sur la création de courbes 3D [8, 24].

Créer des courbes 3D est une tâche difficile. Trois degrés de liberté sont nécessaires en *entrée* alors que les dispositifs habituels sont bidimensionnels. Bien qu'existant depuis de nombreuses années, les souris 3D (SpaceMouse, SpaceBall¹) restent relativement peu utilisées, peut-être du fait d'une certaine difficulté d'apprentissage. Les dispositifs d'entrée 3D basés sur les gestes "en l'air" [Kinect, SoftKinetic, LeapMotion] posent des problèmes de précision et de fatigue pour des utilisations prolongées [22]. En *sortie* se posent les problèmes habituels de représentation des objets tridimensionnels comme l'occlusion ou la difficulté de comparer la taille des objets s'ils sont placés sur des plans différents [40]. C'est d'ailleurs pourquoi les logiciels de CAO fournissent habituellement des vues orthographiques.

Ces difficultés ont inspiré diverses solutions matérielles ou logicielles. Cependant, du fait de la diversité des contextes applicatifs et des approches adoptées, ces travaux tendent à être dispersés dans les publications de plusieurs communautés scientifiques comme SIGCHI (CHI, UIST) ou SIGGRAPH (Eurographics, I3D, GRAPHITE).

Plusieurs taxonomies ont déjà été proposées dans le domaine de l'interaction 3D, notamment celles de Company et al. [10] et Olsen et al. [39], qui sont probablement les plus proches de notre sujet d'étude. Ces deux taxonomies s'intéressent respectivement à l'évolution des techniques de création de formes 3D à partir du tracé et à celles de modélisation par le tracé. Cependant, ces taxonomies plus générales couvrent davantage de travaux que ceux concernant notre sujet d'étude, tout en se limitant à l'interaction par le tracé, alors que d'autres solutions ont été proposées dans le cas des courbes 3D. De même, l'état de l'art de Jankowski [27] présente un ensemble des techniques d'interaction existantes pour les environnements 3D, mais, bien que très complet, celui-ci ne traite pas spécifiquement de l'édition de courbes 3D.

Dans cet article, nous proposons un espace de caractérisation pour la création et l'édition de courbes 3D, qui vise à capturer les spécificités de ce domaine. Contrairement aux précédentes taxonomies, cet espace met en particulier l'accent sur les différentes modalités [38] proposées dans la littérature. Il vise à rassembler et organiser les travaux présentés dans les différentes communautés

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670454>

1. 3DConnexion

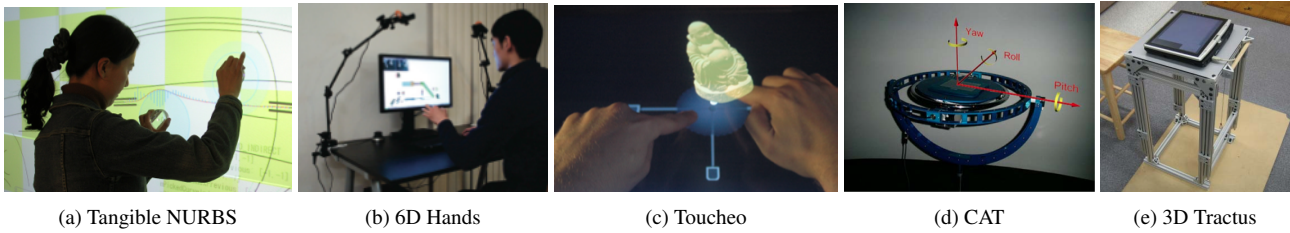


Figure 1: (a) *Tangible NURBS*, un système de création de courbes 3D sur grand écran ; (b) *6D Hands*, un dispositif de CAO via l'interaction dans les airs ; (c) *Toucheo*, un dispositif proposant une interaction novatrice en 2D/3D avec vision stéréoscopique ; (d) *CAT*, un dispositif permettant de se déplacer intuitivement dans une scène 3D ; (e) *3D Tractus*, un dispositif augmentant une tablette avec la profondeur.

académiques citées plus haut. Notre espace de conception s'appuie sur la définition de modalités [38], à travers deux axes : le *dispositif physique* et la *langage d'interaction*. Nous distinguons l'entrée et la sortie pour les dispositifs physiques. Pour le langage d'interaction, nous distinguons la création, l'édition de courbes 3D et la manipulation de la caméra, trois opérations que doivent supporter les techniques d'interaction.

Cet espace vise à la fois à donner une vue globale du domaine et à mettre en évidence plusieurs directions à étudier. Pour cela, nous discutons en particulier un cas d'application de l'édition de courbes 3D : l'édition du son 3D. Ce champ d'étude est particulièrement intéressant car 1) peu connu dans le domaine de l'IHM, 2) en pleine expansion et 3) comporte plusieurs spécificités, comme par exemple, le fait que toutes les dimensions ne sont pas équivalentes du fait des limitations auditives de l'utilisateur. Le son 3D introduit donc de nouveaux défis interactionnels pour la création et l'édition de courbes 3D que nous proposons de discuter dans cet article.

Dans les sections suivantes, nous définissons d'abord la couverture de cet article, puis nous présentons notre espace de caractérisation et les travaux existants. Nous terminons par une discussion sur l'application possible de ces travaux dans le contexte de l'édition de son 3D.

DÉFINITIONS ET COUVERTURE

Définition

Nous définissons une courbe 3D comme une courbe dont les points qui la constituent ont des coordonnées dans un espace à trois dimensions, indépendamment du fait qu'elle soit tracée à l'aide d'un dispositif d'entrée 2D ou 3D. Nous nous intéressons dans cet article aux techniques et dispositifs permettant de créer ou de modifier ces courbes.

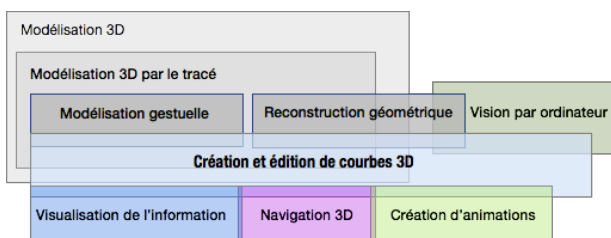


Figure 3: Positionnement de notre champ d'étude

Couverture

La figure 3 montre le positionnement de notre étude, laquelle se situe à la croisée de plusieurs domaines tels que la visualisation de l'information, la navigation 3D, la création d'animations, la vision par ordinateur et la modélisation. A titre d'exemples relatifs à ces domaines, on peut citer l'utilisation de tracés de courbes pour filtrer une visualisation 3D [53, 56], créer des animations [14, 44, 54], définir des trajectoires de caméra dans une scène 3D [34] ou encore créer des routes dans des systèmes de génie civil [36].

Deux champs d'étude sont en lien étroit avec l'édition de courbes 3D : la *modélisation 3D* et, plus spécifiquement, la *reconstruction géométrique* qui est un sous-champ du premier. La *modélisation 3D* vise à construire un ou plusieurs éléments d'une scène en trois dimensions. La plupart des travaux présentés dans cet article relèvent de ce domaine, soit parce qu'ils se basent sur la création de courbes, comme dans le cas de la modélisation par courbes NURBS où la structure du modèle 3D est un maillage de courbes, soit parce que ces travaux présentent des interfaces innovantes pour interagir avec une scène 3D. Signalons cependant que la *modélisation 3D* est un vaste champ d'étude, comprenant d'autres types de travaux que ceux couverts par cet article.

La *reconstruction géométrique* est probablement le champ d'étude le plus proche du sujet qui nous intéresse. Ce sous-domaine de la *modélisation 3D* exploite la vision par ordinateur, pour créer des scènes 3D par interprétation de dessins 2D. À partir du tracé de l'utilisateur, le système est capable d'inférer la forme souhaitée et de la placer correctement dans l'espace 3D, selon le point de vue de l'utilisateur. Bien qu'étant actif, ce domaine est relativement méconnu de la communauté IHM du fait de publications majoritairement dans des conférences comme Eurographics ou I3D. La *reconstruction géométrique* vise une modélisation plus intuitive que les outils de modélisation 3D standard mais a pour inconvénient une moindre précision. Le lecteur pourra se référer aux taxonomies d'Olsen et al. [39] et de Company et al. [10] pour un état de l'art sur le sujet.

Enfin, notre article portant seulement sur la création et l'édition de courbes 3D, nous n'aborderons pas les études dédiées à d'autres méthodes de modélisation telles que la modélisation 3D automatique d'environnements [28].

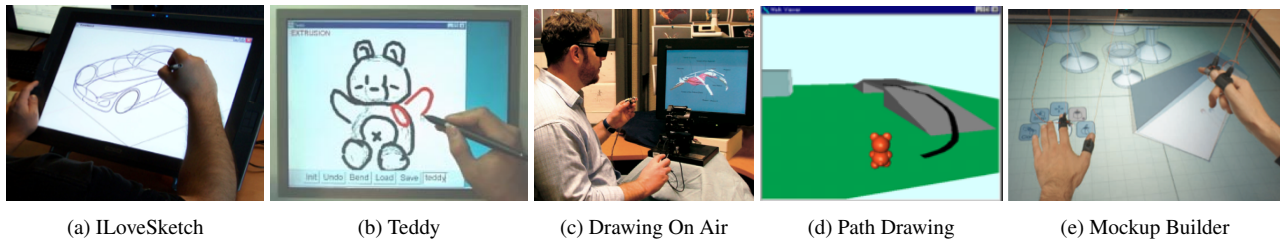


Figure 2: *I Love Sketch*, *Teddy*, des systèmes de modélisation 3D basés sur le dessin 2D ; *Drawing on Air*, un système de modélisation 3D basé sur le retour haptique et l'interaction dans les airs ; *Path Drawing*, une approche de navigation 3D par le tracé 2D ; *Mockup Builder*, un système de modélisation 3D sur table interactive et dans les airs

ESPACE DE CARACTÉRISATION

Les dimensions de notre espace de caractérisation s'appuient sur la notion de *modalité* qui a été utilisée avec succès dans de nombreux états de l'art comme [11, 42, 35]. Nous utilisons la notion de modalité telle que définie par Nigay et al. [38] : une modalité correspond à un couple $\langle D, L \rangle$, où D est un *dispositif physique* et L , un *langage d'interaction*. Un dispositif physique est un élément du système qui acquiert (dispositif d'entrée) ou fournit (dispositif de sortie) des informations à l'utilisateur. Un langage d'interaction est un ensemble d'expressions bien formées et significatives. Nous commençons par examiner les dispositifs, puis nous considérons les langages d'interaction pour la création de courbes 3D, leur édition, et la manipulation de la caméra.

DISPOSITIFS

Nous différencions les dispositifs d'entrée et de sortie.

Dispositifs d'entrée et degrés de liberté

Souris+claviers

La souris (ou le pavé tactile) reste encore le dispositif d'entrée le plus utilisé pour la 3D sur les ordinateurs, sans doute du fait de son faible prix et de son confort d'utilisation. Ce dispositif n'est pourtant pas parfaitement adapté pour effectuer certaines tâches. Comme le font remarquer Ware et al. [52], le placement d'un objet dans une scène 3D requiert 6 degrés de liberté (3 pour les translations et 3 pour les rotations). L'utilisation d'un dispositif aux degrés de liberté réduits, comme une souris, est donc susceptible de rendre l'exécution de la tâche plus longue et plus ardue car l'utilisateur doit constamment changer de mode en faisant des allers-retours entre la scène 3D et les barres de contrôle (barre de menus, barre d'outils).

Plusieurs approches ont été proposées pour pallier cette limitation. Les approches logicielles ont conduit à la proposition de diverses méthodes d'interaction qui sont détaillées dans la section "Langage d'interaction". Sur le plan matériel, on observe que la souris est souvent accompagnée d'un clavier, afin de permettre le changement de modes avec la main gauche sans déplacer le curseur. Par ailleurs, divers dispositifs d'entrée ont été proposés pour manipuler des scènes 3D (et par la même occasion, des courbes 3D), que nous discutons ci-après.

Souris augmentées

Plusieurs souris spécialisées dédiées à l'interaction 3D sont commercialement disponibles, comme la *SpaceMouse* de 3Dconnexion, qui offre 6 degrés de liberté. La *SpaceMouse* se différencie de la *SpaceBall* (qui n'est plus commercialisée), par le fait que la première dispose d'un capteur de type élastique [58] (qui mesure le déplacement en y opposant une force contraire en fonction de la distance), alors que la seconde utilisait un capteur isométrique (qui ne se déplace pas mais mesure la force exercée). Ces souris, qui sont normalement utilisées conjointement avec une souris traditionnelle, nécessitent un certain apprentissage pour pouvoir être utilisées efficacement. Ceci est sans doute dû au fait que, contrairement aux souris habituelles, elles ne sont pas isotoniques et s'apparentent plutôt à une sorte de super joystick. Par opposition, une souris classique mesure le déplacement sans force contraire, ce qui permet d'effectuer du contrôle en position avec une précision arbitraire, en modulant son "gain" de manière adéquate.

Certains prototypes académiques offrent moins de degrés de liberté, mais tentent de faciliter la prise en main. Par exemple, *Rockin' Mouse* [3] offre 4 degrés de liberté (T_x, T_y, R_x, R_y). La translation T_x permet de se déplacer horizontalement dans la scène, T_y verticalement et R_x dans la profondeur, R_y n'étant pas utilisée dans l'application citée. *The Cubic Mouse* [15] est un dispositif en forme de cube, traversé de trois tiges perpendiculaires représentant les trois axes x, y et z. En déplaçant les tiges, l'utilisateur peut exercer un mouvement selon un axe donné. Un capteur à 6 degrés de liberté au centre du cube permet de suivre le dispositif. Enfin, *The Video Mouse* [23] est une souris à 6 degrés de liberté sur un support lui permettant de la pencher en avant, en arrière, à gauche ou à droite, ainsi que de spécifier une rotation sur l'axe vertical.

Surfaces tactiles

Plusieurs travaux pour la 3D s'appuient sur des tablettes ou des tables interactives. Contrairement à la plupart des souris précédentes, ces dispositifs sont conçus pour l'interaction bimanuelle, ce qui permet de partager le contrôle des degrés de liberté entre les deux mains.

Par exemple, *3D Tractus* [33] (Figure 1e) est une tablette montée sur support, dont la hauteur (T_z), est contrôlée par la main gauche. L'utilisateur peut alors dessiner des courbes en 2D (T_x, T_y) sur la tablette avec la main droite.

La métaphore utilisée, est celle d'un volume virtuel, dans lequel la tablette se déplace verticalement. La tablette représente alors le plan sur lequel l'utilisateur dessine et la hauteur du tractus, la position du plan dans le volume. Ce dispositif permet ainsi de contrôler 3 degrés de liberté : T_x et T_y sur la tablette, et T_z en déplaçant la tablette. Enfin, certains systèmes sur tablette (Pressure-based 3D Curve Drawing [31, 32]) n'utilisent que la pression comme entrée supplémentaire pour la profondeur. Ainsi, plus la pression exercée est forte, plus la section de la courbe est épaisse et, par métaphore, proche de l'utilisateur. Cette représentation repose donc sur une analogie entre l'épaisseur du tracé et la profondeur de la courbe dans l'espace 3D.

Des travaux sur les tables interactives, exploitent la technologie multi-points [12]. *Mockup builder* [13] augmente les tables multi-points en permettant d'interagir au dessus de la table, grâce à un câble attaché aux doigts qui permet de capturer leur hauteur. Un avantage de la résistance du câble, est qu'il permet de soutenir le bras et donc de limiter la fatigue, un problème courant de l'interaction "en l'air". Le *CAT* [21] (Figure 1d) est une tablette interactive, dont la surface peut être orientée dans n'importe quelle direction (R_x , R_y et R_z). De plus, des capteurs de pression permettent d'obtenir les trois translations (T_x , T_y et T_z) par contrôle de valeur. Enfin, une tablette au centre du dispositif permet de suivre la position d'un stylo numérique dans une scène 3D. Cette table permet donc à la fois d'interagir en 2D sur la tablette et en 6D avec la table.

Interaction tangible

L'interaction tangible consiste à interagir avec un système numérique à l'aide d'objets physiques fortement couplés aux données numériques manipulées. Peu de véritables systèmes tangibles ont été proposés pour l'interaction 3D. Cependant, nous retenons *ShapeTape*, *ZeroN* et *Surface Drawing*.

ShapeTape [5, 19], qui est illustré Figure 4a, est un ruban physique déformable permettant la modélisation précise d'une courbe, la construction d'une surface, l'extrusion et la construction d'objets de révolution. Il est composé de 32 capteurs optiques, distribués uniformément le long du ruban. Ces capteurs fournissent à chaque position, le degré de courbure (bend) et la torsion (twist), ce qui permet de reconstruire la forme exacte du ruban. Dans une de ses versions, ce dispositif est accompagné d'une double pédale et d'un palet disposés au sol qui permettent de contrôler la caméra avec les pieds, les deux mains étant déjà occupées.

ZeroN [34] (Figure 4b) est un aimant lévitant dans un champ magnétique. La position de cet aimant est capturée avec un algorithme de vision par ordinateur et le système peut déplacer cet aimant en modifiant le champ magnétique. Un vidéo-projecteur permet d'afficher de l'information sur ou autour de l'aimant. En pratique, ce dispositif peut servir à spécifier la trajectoire d'une caméra dans un espace 3D.

Finalement, *Surface Drawing* [45] (Figure 5b) permet de créer des formes 3D, en faisant des gestes "en l'air". Ce système repose principalement sur l'interaction gestuelle

(paragraphe suivant), mais il utilise aussi des outils tangibles complémentaires comme une pince et une gomme.

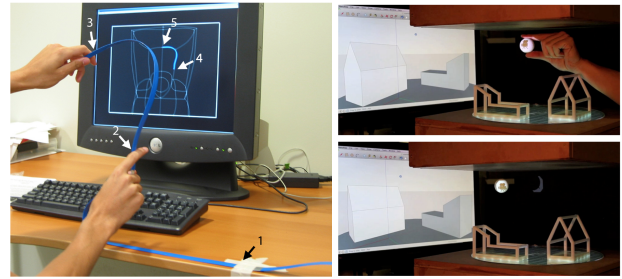


Figure 4: Exemples de dispositifs tangibles. (a) ShapeTape, (b) zeroN

Interaction "en l'air"

L'interaction gestuelle "en l'air" peut sembler en parfaite adéquation avec l'édition de courbes 3D car le nombre de degrés de liberté est le même dans les deux cas. Nous distinguons deux classes de dispositifs : ceux laissant libres les mains de l'utilisateur comme Kinect ou LeapMotion, et ceux nécessitant de tenir un dispositif.

Malgré les avancées de la vision par ordinateur et l'introduction de dispositifs à prix abordables, comme Kinect ou LeapMotion, peu de systèmes permettent véritablement de créer des courbes 3D à mains nues. Nous notons cependant *6D Hands* [50] (Figure 1b), un système de CAO, permettant d'interagir avec des modèles 3D avec les deux mains et sans marqueurs. Le dispositif fonctionne à l'aide de deux caméras Playstation 3 Eye.

Une difficulté récurrente de ce type d'interaction est de déterminer le début et la fin du geste, ce qui impose souvent d'introduire des gestes supplémentaires servant de délimiteurs. En conséquence, davantage de systèmes reposent sur la manipulation d'un objet physique, comme par exemple, un bras haptique Phantom [49] (Figure 2c) (qui fournit également du retour de force), la *Bat* [52], que l'on peut manipuler en l'air grâce à un capteur magnétique Polhemus 3Space Isotrak.

D'autres systèmes ont été conçus pour l'interaction gestuelle bimanuelle en utilisant des gants [45] ou des poignées [2, 4, 17, 18]. Par exemple, des travaux effectués chez Alias|Wavefront [4, 17, 18] (Figure 5a), utilisent des poignées (une dans chaque main), symbolisant les deux extrémités d'un ruban de scotch. Cette technique qui se base sur les techniques de conception de design automobile s'appuie sur une métaphore, celle d'un utilisateur dessinant avec une bande de ruban adhésif, une main collant le scotch progressivement au support, tandis que l'autre tire la bobine et guide le tracé.

Dans tous les cas, ces dispositifs peuvent rapidement entraîner de la fatigue si le bras ne repose pas la plupart du temps sur une surface (syndrome "Gorilla arm") [22]. C'est pourquoi, il est utile de distinguer les dispositifs permettant d'interagir à proximité d'une surface plate, comme avec le bras haptique Phantom ou Leap Motion, des dispositifs ne permettant pas comme [2] (Figure 1a).



Figure 5: Exemples de dispositifs pour l'interaction dans les airs. (a) TapeDrawing, (b) Surface Drawing

Dispositifs de sortie et perception utilisateur

Le retour d'information est capital pour que l'utilisateur comprenne la correspondance entre l'action effectuée via le dispositif d'entrée et son effet sur le système. Nous distinguons les écrans traditionnels, les écrans 3D et les dispositifs offrant du retour haptique.

Ecrans traditionnels

La plupart des systèmes proposés s'appuient sur des écrans de bureau. L'espace écran est souvent divisé en 4 parties (depuis Sketchpad III [29]) dans les logiciels commerciaux : 3 projections orthogonales sur les plans X, Y et Z ainsi qu'une vue en perspective. D'autres systèmes utilisent de larges écrans (ex : [2, 4]) n'affichant qu'une seule vue pour favoriser l'immersion.

Ecrans stéréoscopiques

Les écrans stéréoscopique sont naturellement adaptés à la visualisation et à la manipulation d'objets 3D et peuvent donner la sensation que l'objet se trouve hors de l'écran, à une certaine distance de l'utilisateur comme dans [49]. Ils peuvent aussi être couplés à un suivi de position de la tête permettant à l'utilisateur de voir l'objet sous différents angles sans avoir à exercer un contrôle explicite de la position de la caméra, ce qui libère les mains pour d'autres opérations. *Mockup Builder* [13] (Figure 2e) en est un exemple reposant sur l'utilisation du Kinect.

Toucheo [20] (Figure 1c) renforce la perception qu'une forme est en trois dimensions. Pour faire cela, un écran stéréoscopique placé en haut du dispositif génère une image qui est ensuite réfléchiée sur une surface intermédiaire, via un miroir à mi-chemin de la surface d'interaction. L'image est ainsi formée au dessus de la surface d'interaction et donne une hauteur à l'objet dans l'espace physique. Ce système effectue également un suivi de la position de la tête de l'utilisateur, ce qui permet de renforcer l'effet perceptif.

D'autres systèmes comme *Eraser Pen* [12] ou *Responsive Workbench* [30] emploient des lunettes pour renforcer la sensation 3D, mais ces techniques ont pour inconvénient de forcer l'utilisateur à porter un tel objet.

Retour haptique

Finalement, quelques systèmes renforcent la vision grâce à une modalité sensorielle complémentaire. C'est le cas de *Drawing on Air* [49] (Figure 2c) qui fournit du retour haptique à l'utilisateur via un bras articulé à retour de force Phantom, qui sert aussi de dispositif d'entrée comme indiqué dans la section précédente.

LANGAGE D'INTERACTION

Nous présentons ici, le second axe de notre espace de conception, intitulé langage d'interaction. Il vise à présenter les techniques d'interaction existant avec les dispositifs précédents. Cet axe est organisé en trois sous-parties reflétant les trois types de tâches de l'édition de courbes 3D.

Création de courbes 3D

Pour créer une courbe 3D avec un dispositif à deux degrés de liberté, plusieurs stratégies sont possibles. La première consiste à la créer en deux temps (approche séquentielle). La seconde nécessite que l'utilisateur définisse des contraintes (contraintes) ou que le système les infère automatiquement (inférence). Enfin, la dernière stratégie consiste à transformer la tâche de création en une tâche de sélection (choix). Nous présentons ces différentes stratégies.

Approche séquentielle

L'utilisateur réalise d'abord une courbe sur un plan puis traite la 3ème dimension dans un second temps. Certains systèmes ont proposé d'utiliser l'*ombre portée* [9, 57]. L'utilisateur dessine d'abord la courbe voulue sur le plan X-Y. Il dessine ensuite l'ombre portée de la courbe. A l'aide de la courbe et de son ombre portée, le système infère la profondeur de celle-ci et crée une courbe 3D.

Définition de contraintes

L'utilisateur peut définir lui même ses contraintes pour réaliser une courbe 3D avec un dispositif 2D. Les contraintes peuvent être définies avant le tracé de la courbe. Par exemple, dans *3-Draw* [43], l'utilisateur précise à l'avance le point de départ et le point d'arrivée de la future courbe. Avec *Terrain Sketching* [16], l'utilisateur définit d'abord un volume afin de borner la partie du terrain à remodeler. La définition de contraintes peut également être réalisée en même temps que le tracé grâce à de l'interaction bi-manuelle ou tangible. L'utilisateur peut définir un plan de symétrie avec la main gauche ou d'autres contraintes dans *Mockup Builder* (Figure 7b). Dans *ShapeTape* [5, 19] il utilise un ruban physique pour contraindre le tracé de la courbe.

Inférence

Les contraintes peuvent être inférées automatiquement par le système. Par exemple, *Snap-to-grid*, verrouille des paramètres tels que la définition d'un plan 2D au sein d'une surface 3D pour simplifier la création de courbe selon un axe (ou plan) donné [37, 43]. Cette technique repose sur la métaphore d'un aimant attirant le tracé de l'utilisateur.

Choix

Finalement, la dernière stratégie consiste à transformer la tâche de création de courbes 3D en une tâche de choix d'un élément dans une liste ou un menu. Cette stratégie peut s'avérer très efficace si le nombre de choix possibles est limité. Une approche similaire permet de récupérer des modèles 3D dans une base de données, en utilisant la reconnaissance du tracé [54].

Ces quatre approches sont utiles pour pallier les limitations des dispositifs 2D. Elles peuvent également servir à augmenter la précision du tracé avec des dispositifs 3D.

Édition de courbes 3D

Nous distinguons 3 sous-tâches constituant l'édition de courbes 3D : la sélection, le remodelage et la suppression.

Sélection

Lorsque le dispositif d'entrée est aussi précis qu'une souris, l'utilisateur peut simplement pointer puis cliquer pour sélectionner une courbe. Dans le cas des gestes 3D, le geste de pinch est généralement utilisé pour la sélection comme pour *Mockup Builder* [13] ou *6D Hands* [50] où il fait office de délimiteur [55]. En ce qui concerne l'interaction tangible, la courbe la plus proche de l'objet physique manipulé est automatiquement sélectionnée.

Remodelage

Le remodelage permet d'affiner une section de courbe une fois que celle-ci est sélectionnée. Plusieurs stratégies ont été proposées. La première consiste à déplacer des *points de contrôle*, comme pour les courbes de Bézier. Cette forme d'interaction, très précise, convient en CAO mais pose problème aux graphistes, dont la représentation mentale ne correspond pas au modèle conceptuel de cette technique. Ainsi, en s'appuyant sur les habitudes des dessinateurs professionnels, Baudel et al. [6] ont proposé la notion d'*oversketching* (Figure 6a). L'utilisateur repasse sur son trait plusieurs fois créant ainsi plusieurs tracés. Ceux-ci sont alors fusionnés pour produire une nouvelle courbe qui moyenne les précédentes. Cette technique a ensuite été reprise dans plusieurs travaux [25, 57, 16, 24, 1].

L'interaction tangible est particulièrement adaptée au remodelage car elle facilite la manipulation de la partie de l'objet physique qui intéresse l'utilisateur. Par exemple *ShapeTape* [5, 19] illustre cette possibilité dans le cas d'un ruban physique.

Le lissage est un cas particulier de remodelage (passif) consistant à supprimer le bruit du dispositif d'entrée ou celui lié à l'inexpérience de l'utilisateur. Plusieurs travaux comme *Elasticurves* [48], *Mockup Builder* [13] ou *I LoveSketch* [1] (Figure 2a) offrent cette fonctionnalité, afin de rendre les courbes plus esthétiques.

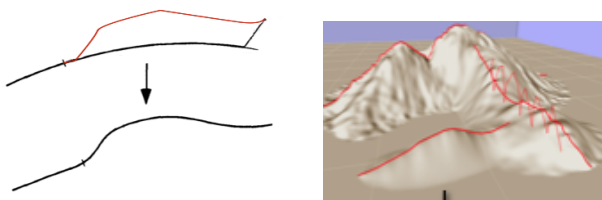


Figure 6: (a) Oversketching, (b) Suppression

Suppression

L'utilisateur est parfois amené à devoir effacer un tronçon de courbe. Il peut par exemple "gribouiller" la partie de la courbe à retirer (Figure 6b) comme dans *3-Draw* [43] ou *I LoveSketch* [1]. Cette technique a été adoptée dans un certain nombre d'autres systèmes comme par exemple *Mockup Builder* ou *Terrain Sketching* [16].

Manipulation de la caméra

La création ou l'édition de courbes 3D peut introduire des problèmes d'occultation par l'objet lui-même (par

exemple s'il comporte une boucle) ou par d'autres objets dans la scène. Une manipulation aisée de la caméra est donc essentielle pour le processus d'édition. Dans *6D Hands* [50], qui, comme vu précédemment, repose sur l'interaction gestuelle en l'air, l'utilisateur peut translater la caméra via une interaction bimanuelle consistant à pincer l'espace avec les deux mains puis tirer pour déplacer la scène. D'autres opérations permettent de faire une rotation de la scène en mimant l'opération de rotation d'une feuille de papier virtuelle avec les deux mains (Figure 7a). Dans [19], un système permettant de créer des courbes 3D à l'aide d'un ruban physique, l'utilisateur peut manipuler la caméra avec des opérations de pan et de zoom, le choix du mode se faisant en fonction de l'orientation du ruban par rapport à l'écran.

D'autres systèmes, comme le *Fish tank VR* [51], utilisent une vue en perspective couplée à la position de la tête (Head coupled perspective [47]) ce qui permet d'adapter la position de la caméra dans le monde 3D en fonction de la position de la tête de l'utilisateur. Le rendu reste ainsi cohérent pour celui-ci, contrairement à une vue standard statique, tout en lui fournissant des indices de perception telles que la profondeur de la scène ainsi que les relations entre objets, d'un point de vue structurel. Cette technique favorise l'immersion [51] en renforçant la perception 3D mais évite surtout des changements de modes fréquents entre le mode "édition" et le mode "manipulation de la caméra". Ceci est alors possible car ces deux tâches sont associées à des dispositifs différents (souris et head-tracking).

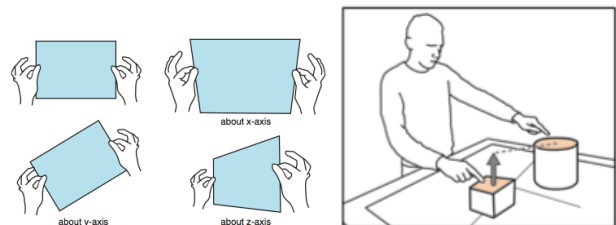


Figure 7: (a) Rotation de la caméra, (b) Définition d'une contrainte sur la hauteur

CAS D'APPLICATION : LE DÉFI DU SON 3D

La création et l'édition de courbes 3D couvrent de nombreux domaines d'applications. Cette diversification explique probablement la variété des techniques d'interaction présentées dans ce papier. Par exemple, plusieurs techniques d'interaction ont été proposées dans le contexte de la navigation 3D [8, 24], la modélisation 3D [57, 49, 43], la création d'animations [54, 44], la génération de terrains [8, 16], la création de routes [36], le design automobile [4, 17, 18, 2]. Nous constatons, cependant, que peu de travaux portent sur le domaine du son 3D.

Définition et défis interactionnels

Le son 3D est un ensemble de techniques permettant d'enregistrer une ou plusieurs sources sonores et de les restituer, indépendamment de leurs positions originales, à une position précise dans l'espace de l'utilisateur. La faible quantité de travaux portant sur l'édition de trajectoires

pour le son 3D s'explique sans doute par le fait que la spatialisation audio est un domaine récent, quoique amené à prendre une importance croissante dans les prochaines années. Comme pour les autres domaines d'applications, le son 3D nécessite la spécification de trajectoires 3D, mais comporte plusieurs spécificités intéressantes.

Tout d'abord, toutes les dimensions ne sont pas équivalentes, car les utilisateurs ne perçoivent pas la direction du son aussi bien sur l'élévation que sur l'azimut. Ceci implique que le besoin de précision diffère selon les directions. D'autre part, comme pour les animations, l'aspect temporel est important, car les sources sonores peuvent se déplacer au cours du temps. L'ajout de cette quatrième dimension temporelle, en plus des trois dimensions spatiales, rend le système d'autant plus complexe et nécessite donc des techniques d'interaction appropriées pour prendre en compte cette difficulté supplémentaire. De plus, l'édition de trajectoires de sources sonores 3D peut être fait par des artistes durant des performances live, ce qui nécessite de pouvoir interagir rapidement, afin de suivre le rythme de la musique, et sans erreur, car il n'est pas envisageable de réaliser plusieurs essais lors d'une performance en public.

Pour ces différentes raisons, nous pensons que le domaine du son 3D est un champ d'étude intéressant, introduisant de nouveaux défis interactionnels pour la création et l'édition de courbes 3D. Nous discutons brièvement les principales méthodes utilisées aujourd'hui dans les applications commerciales, afin de donner un aperçu des techniques d'interaction existantes et quelles tâches celles-ci permettent de réaliser.

Méthodes et logiciels existants

Aujourd'hui, l'édition d'une trajectoire suit généralement les étapes suivantes : création d'une trajectoire de base, transformation ou édition de la trajectoire, et définition d'un mode de lecture pour son exécution. Cette dernière étape est spécifique au son 3D. Plusieurs logiciels existent comme Holo-Edit [7], Hyperspace [41] ou IanniX [26]. Ils utilisent tous la souris comme dispositif d'entrée et l'écran comme dispositif de sortie.

Création de courbes. Plusieurs stratégies sont adoptées pour la création de trajectoires 3D dans ces logiciels. Cela peut être via l'implémentation d'un algorithme ou d'un script comme dans OM-Prisma [46], Holo-Edit [7] et IanniX [26] ou par la sélection de trajectoires prédéfinies comme dans Holo-Edit [7] ou Hyperspace [41]. Cela peut également être fait de manière graphique comme dans Holo-Edit [7], qui propose de positionner à la souris les sources sonores dans l'espace à l'aide d'une visualisation 3D de la scène couplée à une vue en plongée. Hyperspace [41] permet de tracer la position des sources dans un espace bidimensionnel représentant le plan horizontal. Les positions sont représentées sous forme de coordonnées cartésiennes dans un espace 2D. Hyperspace [41], OM-Prisma [46] et IanniX [26] permettent de tracer des fonctions continues à la souris représentant les trajectoires dans un espace bidimensionnel. Une fonction définissant l'élévation dans le temps peut être associée pour obtenir une trajectoire tridimensionnelle.

Edition de courbes. Ces logiciels offrent des méthodes de lissage. Holo-Edit [7] propose également un ensemble de transformations spatiales (expansion, rotation, symétrie, translation, etc.) et temporelles (étirement, ré-échantillonnage, inversion, etc.), pour modifier une trajectoire dans son intégralité, ou partiellement. Comme mentionné plus haut, les artistes peuvent être amenés à éditer les trajectoires pendant un concert. Il est possible d'exécuter un algorithme ou un script préalablement défini, de changer le sens de déplacement de la source sonore, ainsi que d'autres effets. Enfin, Hyperspace [41] propose un mode de contrôle supplémentaire, qui se situe entre création de trajectoire et performance, qui permet de naviguer spatialement entre deux positions tridimensionnelles, préalablement définies via un curseur de défilement.

CONCLUSION

Nous avons présenté un espace de conception pour la création et l'édition de courbes 3D. Bien que ce domaine ait de nombreuses applications, il n'existait pas, à notre connaissance, de revue de littérature ou d'espace de conception centré sur ce thème.

Ce travail a permis de rassembler et d'organiser de nombreux travaux partagés entre la communauté IHM et la communauté graphique, afin de fournir une vue plus globale du domaine. Nous terminons cette étude en évoquant un contexte d'application intéressant, le son 3D, qui a pour l'instant reçu assez peu d'attention, en ce qui concerne la création et l'édition de courbes. La suite de nos travaux visera à transposer et générer de nouvelles techniques d'interaction à partir de cet espace de conception.

REMERCIEMENTS

Le travail présenté dans cet article a été financé par l'agence nationale de la recherche (ANR 13 CORD 008) et réalisé au sein du consortium EDISON 3D.

BIBLIOGRAPHIE

1. Bae S.-H., Balakrishnan R. & Singh K. Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models. In Proc. *UIST '08*, ACM (2008), 151–160.
2. Bae S.-H., Kobayash T., Kijima R. & Kim W.-S. Tangible nurbs-curve manipulation techniques using graspable handles on a large display. In Proc. *UIST '04*, ACM (2004), 81–90.
3. Balakrishnan R., Baudel T., Kurtenbach G. & Fitzmaurice G. The rockin' mouse: Integral 3d manipulation on a plane. In Proc. *CHI '97*, ACM (1997), 311–318.
4. Balakrishnan R., Fitzmaurice G., Kurtenbach G. & Buxton W. Digital tape drawing. In Proc. *UIST '99*, ACM (1999), 161–169.
5. Balakrishnan R., Fitzmaurice G., Kurtenbach G. & Singh K. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In Proc. *I3D '99*, ACM (1999), 111–118.
6. Baudel T. A mark-based interaction paradigm for free-hand drawing. In Proc. *UIST '94*, ACM (1994), 185–192.
7. Cabaud B. & Pottier L. Le contrôle de la spatialisation multi-sources - nouvelles fonctionnalités dans holophon version 2.2. In Proc. *Actes des Journées d'Informatique Musicale* (2002).
8. Cohen J. M., Hughes J. F. & Zeleznik R. C. Harold: A world made of drawings. In Proc. *NPAR '00*, ACM (2000), 83–90.
9. Cohen J. M., Markosian L., Zeleznik R. C., Hughes J. F. & Barzel R. An interface for sketching 3d curves. In Proc. *I3D '99*, ACM (1999), 17–21.

10. Company P., Piquer A. & Contero M. On the evolution of geometrical reconstruction as a core technology to sketch-based modeling. In Proc. *SBM'04*, Eurographics Association (2004), 97–106.
11. Coutrix C., Nigay L., Pasqualetti L. & Renevier P. RAZZLE : de la conception à l'évaluation d'un système mobile et multimodal. In *UBIMOB 2006, Actes des Troisième Journées Francophones : Mobilité et Ubiquité 2006 (Paris, France, 5-8 septembre 2006)* (2006).
12. De Amicis R., Bruno F., Stork A. & Luchi M. L. The eraser pen: a new interaction paradigm for curve sketching in 3D. *Design* (2002).
13. De Araújo B. R., Casiez G. & Jorge J. A. Mockup builder: Direct 3d modeling on and above the surface in a continuous interaction space. In Proc. *GI '12*, Canadian Information Processing Society (2012), 173–180.
14. Fei G., Lee W.-S., Joslin C. & Xin Z. 3d animation creation using space canvases for free-hand drawing. In Proc. *VRCAI '08*, ACM (2008), 12:1–12:6.
15. Fröhlich B. & Plate J. The cubic mouse: A new device for three-dimensional input. In Proc. *CHI '00*, ACM (2000), 526–531.
16. Gain J., Marais P. & Straßer W. Terrain sketching. In Proc. *I3D '09*, ACM (2009), 31–38.
17. Grossman T., Balakrishnan R., Kurtenbach G., Fitzmaurice G., Khan A. & Buxton B. Interaction techniques for 3d modeling on large displays. In Proc. *I3D '01*, ACM (2001), 17–23.
18. Grossman T., Balakrishnan R., Kurtenbach G., Fitzmaurice G., Khan A. & Buxton B. Creating principal 3d curves with digital tape drawing. In Proc. *CHI '02*, ACM (2002), 121–128.
19. Grossman T., Balakrishnan R. & Singh K. An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In Proc. *CHI '03*, ACM (2003), 185–192.
20. Hachet M., Bossavit B., Cohé A. & de la Rivière J.-B. Toucheo: Multitouch and stereo combined in a seamless workspace. In Proc. *UIST '11*, ACM (2011), 587–592.
21. Hachet M., Guitton P. & Reuter P. The cat for efficient 2d and 3d interaction as an alternative to mouse adaptations. In Proc. *VRST '03*, ACM (2003), 225–112.
22. Hincapié-Ramos J. D., Guo X., Moghadasian P. & Irani P. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In Proc. *CHI '14*, ACM (2014), 1063–1072.
23. Hinckley K., Sinclair M., Hanson E., Szeliski R. & Conway M. The videomouse: A camera-based multi-degree-of-freedom input device. In Proc. *UIST '99*, ACM (1999), 103–112.
24. Igarashi T., Kadobayashi R., Mase K. & Tanaka H. Path drawing for 3d walkthrough. In Proc. *UIST '98*, ACM (1998), 173–174.
25. Igarashi T., Matsuoka S. & Tanaka H. Teddy: A sketching interface for 3d freeform design. In Proc. *SIGGRAPH '99*, ACM Press/Addison-Wesley Publishing Co. (1999), 409–416.
26. Jacquemin G., Coduys T. & Ranc M. Iannix 0.8. In *Actes des Journées d'Informatique Musicale* (2012).
27. Jankowski J. & Hachet M. A Survey of Interaction Techniques for Interactive 3D Environments. In *Eurographics 2013 - STAR* (2013).
28. Jiang N., Tan P. & Cheong L.-F. Symmetric architecture modeling with a single image. In Proc. *SIGGRAPH Asia '09*, ACM (2009), 113:1–113:8.
29. Johnson T. E. Sketchpad III: A computer program for drawing in three dimensions. In Proc. *AFIPS '63 (Spring)*, ACM (1963), 347–353.
30. Krueger W. & Froehlich B. The responsive workbench. *IEEE Comput. Graph. Appl.* 14, 3 (1994), 12–15.
31. Lai C.-Y. & Zakaria N. Pressure-based 3d curve drawing. In Proc. *SG'10*, Springer-Verlag (2010), 156–159.
32. Lai C.-Y. & Zakaria N. Towards an intelligent framework for pressure-based 3d curve drawing. In *Smart Graphics*, M. Christie and T.-Y. Li, Eds., vol. 8698 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, 63–72.
33. Lapiès P., Sharlin E., Sousa M. C. & Streit L. The 3d tractus: A three-dimensional drawing board. In Proc. *TABLETOP '06*, IEEE Computer Society (2006), 169–176.
34. Lee J., Post R. & Ishii H. Zeron: Mid-air tangible interaction enabled by computer controlled magnetic levitation. In Proc. *UIST '11*, ACM (2011), 327–336.
35. Malacria S. & Lecolinet E. Espace de caractérisation du stylo numérique. In Proc. *IHM '08*, ACM (2008), 177–184.
36. McCrae J. & Singh K. Sketch-based path design. In Proc. *GI '09*, Canadian Information Processing Society (2009), 95–102.
37. Michalik P., Kim D. H. & Bruderlin B. D. Sketch- and constraint-based design of b-spline surfaces. In Proc. *SMA '02*, ACM (2002), 297–304.
38. Nigay L. & Coutaz J. Espaces conceptuels pour l'interaction multimédia et multimodale. *Technique et science informatiques* 15, 9 (1996).
39. Olsen L., Samavati F. & Sousa M. C. A taxonomy of modeling techniques using sketch-based interfaces. *Eurographics State of the Art Report* (2008).
40. Ortega M. 3d object position using automatic viewpoint transitions. In Proc. *CHI '13*, ACM (2013), 193–196.
41. Parmerud Å. Hyperspace.
42. Roudaut A. & Lecolinet E. Un espace de classification pour l'interaction sur dispositifs mobiles. In Proc. *IHM '07*, ACM (2007), 99–106.
43. Sachs E., Roberts A. & Stoops D. 3draww: A tool for designing 3d shapes. *IEEE Comput. Graph. Appl.* 11, 6 (1991), 18–26.
44. Santosa S., Chevalier F., Balakrishnan R. & Singh K. Direct space-time trajectory control for visual media editing. In Proc. *CHI '13*, ACM (2013), 1149–1158.
45. Schkolne S., Pruett M. & Schröder P. Surface drawing: Creating organic 3d shapes with the hand and tangible tools. In Proc. *CHI '01*, ACM (2001), 261–268.
46. Schumacher M. & Bresson J. Compositional control of periphonic sound spatialization. In *2nd International Symposium on Ambisonics and Spherical Acoustics* (2010).
47. Sutherland I. E. A head-mounted three dimensional display. In Proc. *AFIPS '68 (Fall, part I)*, ACM (1968), 757–764.
48. Thiel Y., Singh K. & Balakrishnan R. Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2d curves. In Proc. *UIST '11*, ACM (2011), 383–392.
49. Wang C. C. Drawing on air: Input techniques for controlled 3d line illustration. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007), 1067–1081.
50. Wang R., Paris S. & Popović J. 6d hands: Markerless hand-tracking for computer aided design. In Proc. *UIST '11*, ACM (2011), 549–558.
51. Ware C., Arthur K. & Booth K. S. Fish tank virtual reality. In Proc. *CHI '93*, ACM (1993), 37–42.
52. Ware C. & Jessome D. R. Using the bat: A six dimensional mouse for object placement. In *Proceedings on Graphics Interface '88*, Canadian Information Processing Society (1988), 119–124.
53. Wattenberg M. Sketching a graph to query a time-series database. In Proc. *CHI EA '01*, ACM (2001), 381–382.
54. Wilches D., Figueroa P., Conde A. & Samavati F. Sketch-Based Interface for animation for non-experts. *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En* (2012).
55. Wilson A. D. Robust computer vision-based detection of pinching for one and two-handed gesture input. In Proc. *UIST '06*, ACM (2006), 255–258.
56. Yu L., Efstathiou K., Isenberg P. & Isenberg T. Efficient structure-aware selection techniques for 3d point cloud visualizations with 2dof input. *Visualization and Computer Graphics, IEEE Transactions on* 18, 12 (2012), 2245–2254.
57. Zeleznik R. C., Herndon K. P. & Hughes J. F. Sketch: An interface for sketching 3d scenes. In Proc. *SIGGRAPH '06*, ACM (2006).
58. Zhai S. *Human performance in six degree of freedom input control*. PhD thesis, University of Toronto, 1995.

LinkWave : une Liste d'Adjacence Visuelle Interactive pour explorer les Réseaux Pondérés Dynamiques

Nathalie Henry Riche
Microsoft Research
Redmond, WA, USA
nath@microsoft.com

Sheelagh Carpendale
University of Calgary
Calgary, Canada
sheelagh@ucalgary.ca

Yann Riche
Microsoft Surface
Redmond, WA, USA
yannr@microsoft.com

Tara Madhyastha
University of Washington
Seattle, WA, USA
madhyt@u.washington.edu

Nicolas Roussel
INRIA
Villeneuve d'Ascq, France
nicolas.roussel@inria.fr

Thomas J. Grabowski
University of Washington
Seattle, WA, USA
tgrabow@u.washington.edu

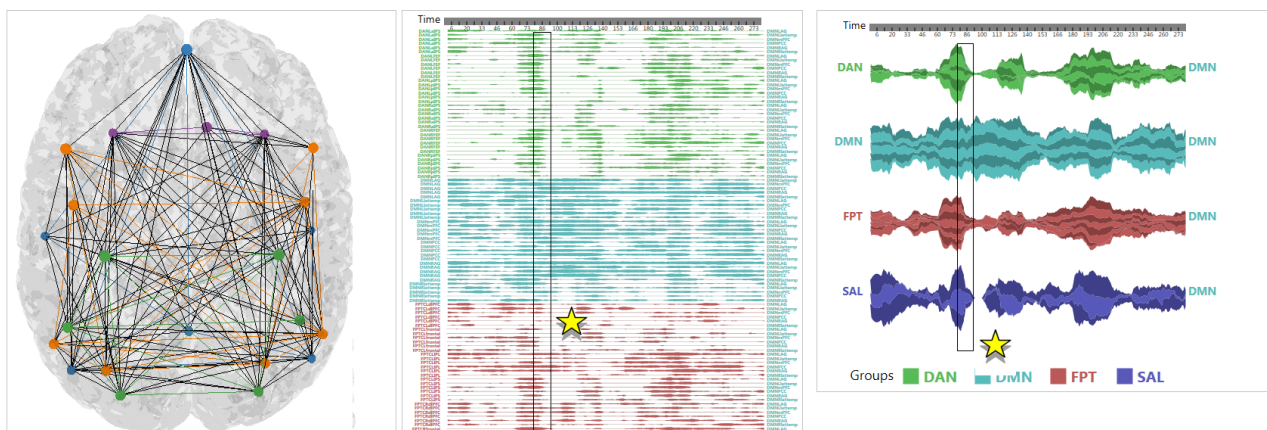


Figure 1: Représentation de la connectivité de 23 régions du cerveau regroupées en 4 groupes fonctionnels lors d'un scan de 5 minutes (300 points). Le diagramme noeuds-liens (à gauche) représente la connectivité à un seul instant donné. La liste d'adjacence visuelle (au milieu et à droite) que nous proposons représente l'évolution temporelle des poids de chacun des liens du réseau. Ces liens peuvent être agrégés pour permettre la découverte de motifs de connexion entre groupes de noeuds. Cette représentation est au coeur du système LinkWave, permettant aux utilisateurs d'explorer interactivement des ensembles de liens. Les étoiles indiquent une période où le poids d'un grand nombre de liens a diminué. La vue agrégée permet de discerner que le poids des liens colorés en turquoise (liens du groupe DMN) n'ont pas suivi ce motif.

RÉSUMÉ

Nous présentons LinkWave, un système pour visualiser interactivement des réseaux pondérés dynamiques. LinkWave est basé sur le concept simple d'une liste visuelle de liens qui peuvent être triés, filtrés et agrégés. LinkWave a été développé en collaborations avec des chercheurs en neurosciences pour analyser les réseaux de connectivité du cerveau issus de données IRMf. Nous expliquons la démarche de conception du système et rendons compte des commentaires des neuroscientifiques impliqués et des premières découvertes réalisées avec le système.

Mots Clés

IRMf ; connectivité fonctionnelle cérébrale ; réseaux pondérés dynamiques ; visualisation d'information

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous.

INTRODUCTION

La recherche en neurosciences s'intéresse de près aux mécanismes cognitifs et neuraux qui sous-tendent les processus complexes de la pensée et du raisonnement^{1, 2}. Une avancée majeure dans ce domaine est l'*imagerie par résonance magnétique fonctionnelle* (IRMf) qui permet d'étudier *in vivo* de manière indirecte l'activité cérébrale. De nombreux chercheurs utilisent les données d'IRMf pour étudier les interactions entre différentes régions du cerveau [29]. Ils modélisent pour cela l'activité cérébrale sous forme de réseaux complexes, les noeuds de ces réseaux correspondant aux régions du cerveau et des liens pondérés dynamiques indiquant la force d'interaction entre deux régions à un instant donné. Les données d'IRMf sont cependant à la fois complexes et volumineuses, ce qui rend leur analyse difficile.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ème} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670461>

1. <http://www.nih.gov/science/brain/>
2. <https://www.humanbrainproject.eu>

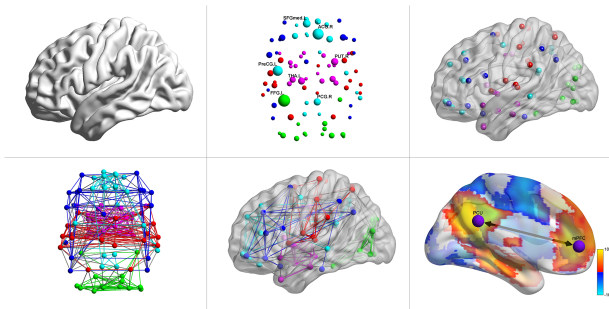


Figure 2: Le fonctionnement du cerveau représenté par des diagrammes nœuds-liens entre régions. Six représentations 3D générées par *BrainNet Viewer* /citebrainnetviewer

L'identification de motifs dans ces réseaux dynamiques est un enjeu important pour la compréhension générale du cerveau et pour l'identification et la compréhension de phénomènes particuliers tels que la maladie d'Alzheimer. Or, cette tâche s'avère extrêmement difficile de par sa nature exploratoire lorsqu'on se limite aux outils d'analyse statistique classiques tels que l'environnement R. Quelques outils de visualisation tel que *BrainNet Viewer* [33] proposent de représenter les interactions déterminées à partir des IRMf en utilisant la représentation la plus commune des réseaux : les diagrammes nœuds-liens. Le placement des nœuds respecte généralement la morphologie du cerveau, chaque nœud étant virtuellement positionné sur la région qu'il représente. Cependant, les représentations nœuds-liens sont généralement difficile à lire pour ces réseaux denses [18], et ce placement diminue plus encore la lisibilité puisqu'il ne permet pas de minimiser le croisement des liens (Figure 2).

De plus, l'aspect dynamique des réseaux est difficile à représenter, les approches les plus courantes consistant à calculer une moyenne des poids de connexion dans le temps, ou à générer une animation omettant éventuellement les liens de poids inférieur à un seuil pour diminuer l'encombrement visuel. S'ils offrent une vue d'ensemble des connexions fonctionnelles, les outils de visualisation existants ne permettent alors pas aux scientifiques d'analyser l'évolution des poids de connexion dans le temps et d'en extraire des motifs. Or, ces tâches sont critiques pour les chercheurs qui tentent de comprendre l'impact des maladies dégénératives telles que Alzheimer ou Parkinson. Une des hypothèses concernant ces maladies est en effet qu'elles dégradent l'interaction entre certaines régions du cerveau, ce qui pourrait se traduire par une modification de certains poids de connexion ou de leur séquençage dans le temps.

Dans cet article, nous décrivons un système d'exploration visuelle destiné à faciliter l'identification de motifs dans les réseaux dynamiques pondérés, appliqué ici au réseaux formés par les zones d'activité du cerveau humain et leurs connexions. Ce système, *LinkWave*, repose sur une représentation qui met en avant l'évolution des poids des liens du réseau dans le temps et les représentent comme des séries temporelles (Figure 1). *LinkWave* offre un nouveau paradigme d'exploration des réseaux dynamiques dans lequel l'utilisateur interagit principalement avec les liens et

non les nœuds du réseau. Nous offrons des interactions simples qui permettent de trier, agréger et comparer ces liens les uns aux autres pour extraire des motifs temporels. Nous expliquons la démarche de conception du système de plus de deux ans avec des chercheurs en neurosciences de l'Université de Washington. Nous rendons également compte des premières découvertes réalisées par ces chercheurs avec *LinkWave* concernant des motifs liés à l'âge du sujet ou à des maladies dégénératives telles que les maladies de Parkinson ou d'Alzheimer.

ANALYSE DES BESOINS

Pour informer la conception de *LinkWave*, nous avons formé une collaboration avec l'équipe de neuroscience Integrated Brain Imaging Center (IBIC) de l'Université de Washington. Ces chercheurs étudient les dégradations des connexions dans le cerveau humain dues à certaines maladies dégénératives comme la maladie de Parkinson ou d'Alzheimer, issues de traumatismes ou de chirurgies, ou simplement dues au vieillissement naturel. Ils analysent des données IRMf comparant les différents motifs de connexion extraits de leurs sujets à un groupe contrôle. Après une série d'entretiens avec six neuroscientifiques (rencontres bi-mensuelles, de deux heures environ, pendant six mois) et l'observation de leurs pratiques existantes d'analyse avec les outils statistiques et visuels traditionnels, nous avons formalisé une liste de tâches typiquement réalisées sur les graphes pondérés dynamiques produits à partir d'IRMf.

Caractéristiques des Données

Les données IRMf mesurent l'activité des régions cérébrales de manière indirecte, via leur consommation d'oxygène. Les neurologues nettoient ces données pour éliminer les bruits provenant des éventuels mouvements du sujet pendant l'acquisition et d'autres facteurs physiologiques tels que la respiration. Dû à la présence de fluide cérébrospinal et à des parties de matière blanche présents dans les images IRMf mais qui ne représentent pas l'activité neuronale, les valeurs des signaux brutes à un moment donné ne sont pas comparables aux valeurs brutes à un autre moment dans le temps. L'information que les scientifiques utilisent est donc toujours une information sur le niveau d'activation d'une région par rapport aux autres à un moment donné. A cet effet, nous calculons des corrélations de Pearson entre régions d'intérêt à différents points dans le temps (à noter que d'autres méthodes existent incluant l'utilisation de modèles statistiques). Le résultat de ces calculs est un réseau pondéré dynamique dont il est généralement supposé que la dynamique au cours du temps reflète le fonctionnement du cerveau.

Les réseaux produits possèdent très peu de nœuds, d'une vingtaine à quelques centaines, chaque nœud pouvant appartenir à un groupe fonctionnel. Par exemple, les réseaux que nos collaborateurs analysent pour identifier des motifs associés à la maladie de Parkinson sont constitués de 23 nœuds faisant partie de 4 groupes fonctionnels différents. Ces groupes représentent les régions qui interagissent ensemble pour accomplir une certaine fonction. Par exemple le *Default Mode Network* (abrégé DMN et coloré en turquoise sur les figures) est un groupe de régions qui interagit lorsque le cerveau est au repos. Ce groupe est un

des premiers affecté chez les sujets atteints de la maladie de Parkinson. Il est utile de noter que le terme “réseau” (“network” en anglais) utilisé en neurosciences ne représente pas le réseau complet mais un groupe de nœuds. Les groupes fonctionnels résultent de nombreux travaux de recherche et sont identifiés par les neurologues a priori.

Le réseau présente une forte densité de liens à chaque instant donné. Le réseau agrégé sur l'intervalle de temps total est un graphe complet (chaque nœud est connecté à chacun des autres nœuds à un moment donné). Cette forte densité est source d'occultation dans les diagrammes nœuds-liens, ce qui nécessite l'utilisation de représentations alternatives telles que les matrices d'adjacence. Le nombre d'instantanés considérés est relativement important, de l'ordre de 300. Ce nombre rend l'utilisation de “*small multiples*” en pratique très difficile.

Formalisation des Tâches

A partir de ces entretiens, nous formalisons une liste de 7 tâches typiquement réalisées sur les graphes pondérés dynamiques extraits des données IRMF de nos collaborateurs.

T1. Caractériser la connectivité entre deux nœuds

Cette tâche consiste à caractériser l'évolution dans le temps de la force de connectivité entre deux nœuds. Il est par exemple important d'identifier les augmentations de poids d'un lien et de considérer leur intensité et leur fréquence.

T2. Caractériser la connectivité d'un nœud

Cette tâche consiste à caractériser l'évolution dans le temps de la connectivité d'un nœud. Il est par exemple important d'identifier un groupe de nœuds auxquels celui-ci est fortement connecté en moyenne, et d'identifier si ces nœuds varient à certains moments.

T3. Identifier des groupes de nœuds similaires

Cette tâche consiste à identifier des nœuds qui présentent des motifs de connexions similaires. Ces nœuds peuvent être considérés similaires de par leur connectivité à des mêmes nœuds et/ou en rapport à l'évolution des poids de leurs liens.

T4. Identifier des cas particuliers

Cette tâche consiste à identifier des nœuds (ou des liens) dont les motifs de connectivité diffèrent des autres. Il est important d'identifier si ces exceptions ont trait à la topologie (différents voisins), à l'évolution des poids de connexion (poids différents du groupe), ou sont dus au séquençement (délai de connexion par rapport au groupe).

T5. Caractériser la connectivité entre plusieurs groupes

Cette tâche consiste à caractériser la force de connexion entre des groupes de nœuds. Identifier les nœuds qui jouent un rôle particulier pour ces groupes, un nœud pivot par lequel passeraient la majorité des connexions entre deux groupes par exemple. Évaluer l'évolution temporelle de ces rôles est également important.

T6. Identifier des états du graphe

Cette tâche consiste à identifier des motifs de haut niveau qui permettent de caractériser un état du réseau. Par

exemple, un état extrême (et peu courant) est un réseau complet, où tous les nœuds sont connectés, ou bien un réseau déconnecté, où aucune connexion n'apparaît. En réalité, ces états sont bien plus nuancés. Ils peuvent être définis par un intervalle de temps dans lequel les nœuds du réseau semblent suivre un motif de connexion particulier. Idéalement, ces états peuvent se répéter dans le temps et il est important de pouvoir les dissocier les uns des autres.

T7. Comparer plusieurs graphes pondérés dynamiques

Étant donné que la structure anatomique du cerveau varie d'un individu à un autre, on peut estimer que la structure fonctionnelle du cerveau va également présenter des variations entre individus. Il est important de pouvoir comparer les motifs entre individus et d'identifier les motifs communs ainsi que leurs variations.

LINKWAVE

Pour répondre aux besoins de nos collaborateurs en neurosciences, nous avons conçu une visualisation interactive qui permet d'accomplir les tâches décrites ci-dessus. Les travaux de recherche sur la visualisation de graphes s'intéressent essentiellement à des tâches centrées sur les nœuds et groupes de nœuds [23]. À l'inverse, les tâches identifiées pour les neuroscientifiques mettent en avant les liens du réseau, et non ses nœuds. Si peu de travaux en visualisation se sont concentrés sur ces questions, elles sont pourtant importantes pour l'analyse de réseaux en général et applicables à d'autres domaines. Par exemple, les chercheurs en sciences humaines et sociales accomplissent des tâches similaires lorsqu'ils analysent les communications par courriers électroniques ou des réseaux sociaux. Dans ces réseaux la force de connexion peut être liée au nombre de messages échangés et identifier les motifs temporels (la fréquence et le rythme général de communication) est crucial à l'analyse des comportements sociaux.

Aider les utilisateurs à accomplir les tâches ci-dessus nécessite un changement de perspective dans lequel *les liens du réseau deviennent les éléments centraux de la représentation visuelle*. Par exemple, analyser l'évolution de la connexion entre deux groupes de nœuds revient à analyser l'évolution des poids des liens connectant les deux

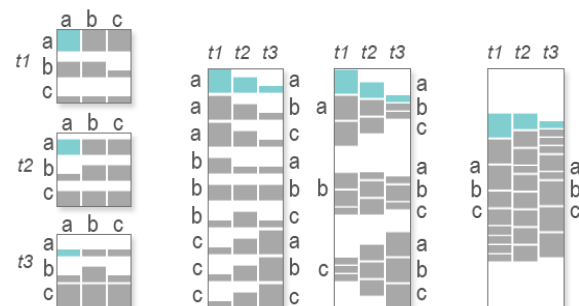


Figure 3: De gauche à droite : matrices d'adjacence pour représenter l'évolution d'un réseau pondéré à trois instants donnés, liste d'adjacence visuelle dans laquelle chaque ligne représente l'évolution des poids d'un lien, agrégation de cette liste illustrant la connectivité de chaque nœud, agrégation de cette liste illustrant la connectivité totale du réseau.

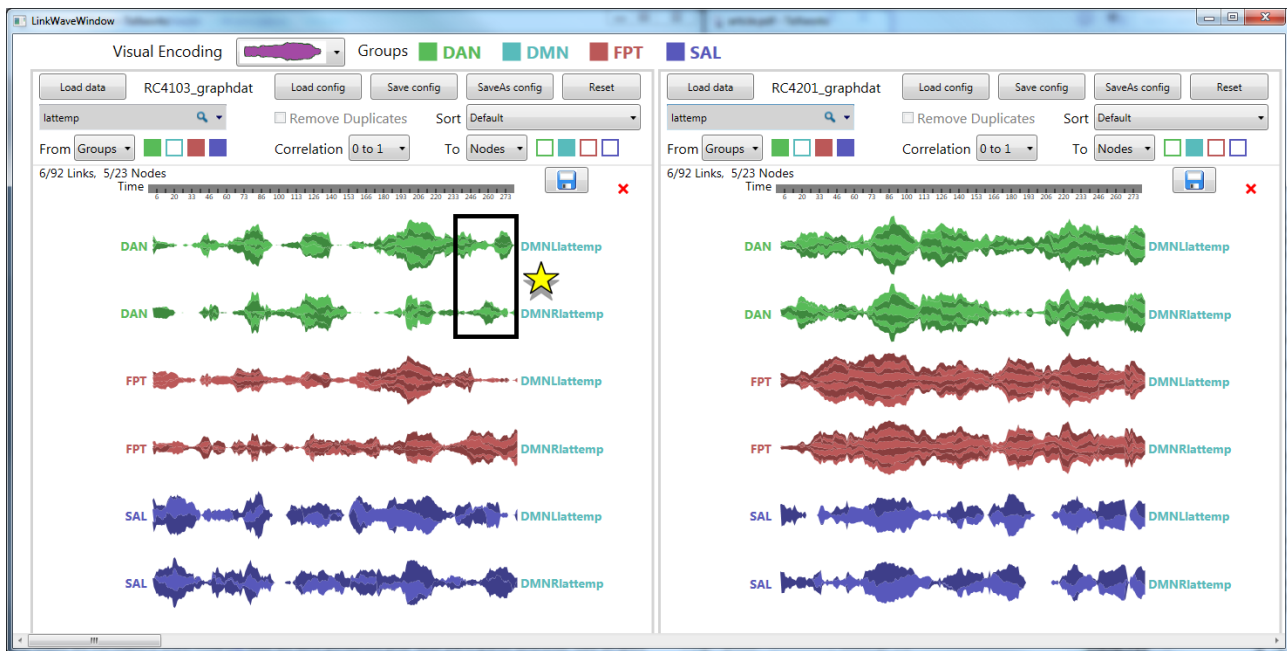


Figure 4: Capture d'écran de la fenêtre principale de LinkWave lors de l'exploration de la connectivité d'un sujet atteint de la maladie de Parkinson (à gauche) et d'un sujet du groupe contrôle (à droite). Les liens vers la région latérale temporale du groupe DMN (lattemp) sont agrégés par leurs sources (régions des groupes DAN, FPT et SAL). Chaque paire de lien d'une couleur représente la connexion à la région lattemp de l'hémisphère droit (DMNRlattemp) et gauche (DMNLlattemp). Alors que ces liens ont des motifs similaires chez le sujet contrôle, la symétrie est altérée chez le sujet avec Parkinson. L'étoile indique une de ces différences.

groupes. LinkWave est une visualisation interactive permettant d'explorer ce nouveau paradigme d'exploration des réseaux dynamiques. Le principe de cette visualisation est illustré par la Figure 3. Il consiste à représenter le réseau dynamique complet par une liste d'adjacence : une table dans laquelle chaque ligne représente un lien du réseau et chaque colonne un point dans le temps. Une cellule dans cette table représente le poids porté par un lien (ligne) à un instant donné (colonne). De fait, une ligne de la table encode l'évolution temporelle de la force de connexion entre deux nœuds. La représentation de l'évolution temporelle des poids des liens dans une représentation unique rend possible la comparaison de plusieurs liens et permet de comparer les rythmes de leur augmentations ou réductions de poids. Bien que les listes d'adjacence soient couramment utilisées pour décrire les données brutes des réseaux, à notre connaissance, ces structures n'ont jamais été utilisées comme outils de visualisation et d'exploration.

Représentation visuelle

LinkWave présente l'évolution des liens pondérés du réseau sous forme de liste. Il s'agit donc de représenter visuellement plusieurs séries de données temporelles. La littérature sur ce sujet est abondante et plusieurs représentations sont possibles. Les représentations les plus communes sont celles à base de lignes (reliant les points), de barres (une par point) ou d'aires (en partie délimitées par les points). Considérant que les poids des liens sont le résultat d'un calcul de corrélation sur une fenêtre glissante, une représentation lissant les changements pour indiquer une tendance d'évolution plutôt que des valeurs précises

semble préférable. Nous nous sommes donc intéressés aux représentations à base de lignes et d'aires. Considérant le grand nombre d'éléments à représenter et la prédisposition humaine à percevoir des motifs symétriques [14], nous nous sommes intéressés aux graphiques d'aires symétriques. Nous avons également exploré des représentations plus compactes tels que les heatmap et graphes d'horizons, mais celles-ci ont été critiquées par nos utilisateurs en raison de la difficulté à discerner les variations de couleur et d'opacité et, plus généralement, à identifier des motifs saillants.

Etant donné que chaque série temporelle représente l'évolution du poids d'un lien du réseau, un nœud source (placé à gauche de l'aire) et un nœud destination (placé à droite) y sont donc associés. Ces informations topologiques peuvent être utilisées pour agréger les liens et révéler la connectivité à un plus haut niveau d'abstraction. Par exemple, agréger les liens qui ont le même nœud source permet de visualiser la connectivité générale de chacun des nœuds du réseau.

Dans les données visualisées dans la Figure 4, nous employons seulement trois niveaux d'agrégation possibles pour la source et/ou la destination des liens : un nœud, un groupe de nœud, ou tous les nœuds du réseau. Cela revient à attacher une hiérarchie de trois niveaux à droite et cette même hiérarchie à gauche, résultant en 9 combinaisons (nœud-nœud ; nœud-groupe ; nœud-tous ; groupe-nœud ; groupe-groupe ; groupe-tous ; tous-nœud ; tous-groupe ; tous-tous). Ces agrégations peuvent être généralisées à une hiérarchie de profondeur arbitraire. Pour représenter ces agrégations, nous utilisons des *stream-*

graphs. Nous n'avons pas choisi cette représentation pour permettre une analyse détaillée des diverses couches, mais pour leur effet visuel qui permet de rappeler à l'utilisateur qu'il s'agit d'une agrégation de liens et non d'un objet unique.

Interactions

Les liens du réseau sont exponentiellement plus nombreux que les noeuds. En effet, un réseau complet de n noeuds comporte $n \times (n - 1) \div 2$ liens non dirigés. Le plus petit réseau que nous avons étudié comporte 23 noeuds et 253 liens évoluant sur 280 points dans le temps. Le principal défi est donc de proposer des techniques d'interaction permettant le passage à l'échelle de la visualisation et son utilisation sur un (ou deux) écran(s) d'ordinateur standard(s). LinkWave offre aux utilisateurs des interactions standards sur les données temporelles tels que zoomer, trier, filtrer et agréger. Ces interactions ont été également adaptées pour explorer la topologie du réseau et permettre aux utilisateurs de spécifier un ensemble particulier de liens.

LinkWave permet le tri des séries temporelles selon des valeurs à des instants spécifiques ou à partir de mesures statistiques simples (minimum, maximum, médiane). Le tri peut être également spécifique à la topologie du réseau, en réordonnant la liste alphabétiquement par les noms des noeuds ou des groupes source ou destination. Le filtrage est également adapté à l'exploration de réseaux. Il est par exemple possible d'explorer la connectivité intra-groupe (en filtrant les autres groupes en sources et destinations des liens), ainsi que la connectivité inter-groupes (en spécifiant un groupe donné en source et un groupe différent en destination). La visualisation interactive du réseau s'effectue donc en interagissant avec différents ensembles de liens au travers du panneau de contrôle situé en haut de la fenêtre LinkWave (Figure 4).

Scenario d'utilisation

Pour illustrer le type de découvertes que permet LinkWave, nous présentons un court scénario d'utilisation. Emma souhaite explorer les données issues de son étude avec des sujets atteints de la maladie de Parkinson. Elle charge un premier sujet et isole les régions du groupe fonctionnel DMN (régions actives à l'état de repos du sujet), que la littérature scientifique désigne comme particulièrement affecté par la maladie. Elle utilise les boutons de filtrage de LinkWave pour isoler les 117 liens (sur un total de 506) dont la destination est une région du groupe DMN. Elle effectue ensuite un tri par le poids moyen de chaque lien pour avoir un aperçu des connexions les plus fortes chez ce sujet. Elle note que les 9 liens dont la moyenne des poids est la plus élevée sont des liens internes de DMN (la source et la destination sont des régions de DMN). Intriguée, elle se demande si le tri serait similaire chez un sujet du groupe contrôle. Elle copie la configuration de la vue d'un clic et charge les données d'un sujet contrôle. Cette fois, seuls 3 liens internes du groupe DMN apparaissent dans le top 10 (Figure 5).

Encouragée par la confirmation d'une différence entre les motifs de connexion intra-groupe DMN, Emma décide d'explorer s'il existe une différence entre les liens

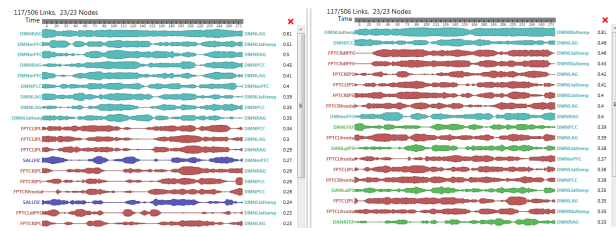


Figure 5: Listes de liens pondérés ordonnée par le poids moyen pour un sujet atteint de Parkinson (à gauche) et un sujet contrôle (à droite). Les liens internes au groupe DMN sont colorés en bleu.

inter-groupes. Elle filtre les sources des liens qui appartiennent au groupe DMN pour éliminer les liens intra-groupe. Elle agrège ensuite les noeuds sources des 102 liens restant par leurs 3 groupes respectifs (FPT, DAN et SAL). En étudiant l'évolution de ces 18 liens agrégés, elle identifie des points de synchronisation particulièrement salients dans le sujet avec Parkinson. Emma note également que certains liens semblent avoir un motif différent des autres. Il s'agit des liens avec la région latérale temporale (lattemp) au sein du groupe DMN. Elle effectue une recherche sur ce libellé pour ne garder que les 6 liens correspondants chez les deux sujets. Chaque paire de liens représente les connexions agrégées des régions d'un groupe fonctionnel à la région lattemp dans l'hémisphère droit (DMNRIlattemp) et gauche (DMNLIlattemp) du cerveau. Le sujet contrôle exhibe des motifs similaires pour chacune de ces paires alors que cette symétrie est altérée chez le sujet avec Parkinson (Figure 4). Cette découverte semble confirmer l'hypothèse que certaines régions du cerveau se désynchronisent due à la maladie de Parkinson. Cependant, bien d'autres explorations et analyses statistiques seront nécessaires pour accepter ou rejeter définitivement cette hypothèse.

RETOUR D'EXPÉRIENCE

Méthodologie

Afin d'itérer sur l'utilisabilité de LinkWave et de comprendre si ce système peut permettre de faire des découvertes lors de l'exploration d'IRMF, nous avons suivi la méthode d'évaluation longitudinale MILCS proposée par Shneiderman et Plaisant en 2006 [28]. Cette méthode ethnographique consiste à collaborer étroitement avec une poignée d'utilisateurs potentiels dans les conditions les plus réalistes possibles (utilisant leurs données et leur cadre de travail). LinkWave a été utilisé pendant trois mois par deux chercheurs de l'Université de Washington (occasionnellement par leurs étudiants). Les scientifiques ont installé une copie du logiciel et ont conduit leurs explorations seuls. Nous leurs avons demandé de documenter leurs découvertes en faisant des captures d'écran et en utilisant les points de sauvegarde du logiciel. Enfin, deux fois par mois, pendant deux heures environ, nous avons discuté avec eux de leurs explorations faites avec LinkWave, des éventuels problèmes rencontrés et des évolutions possibles de notre système.

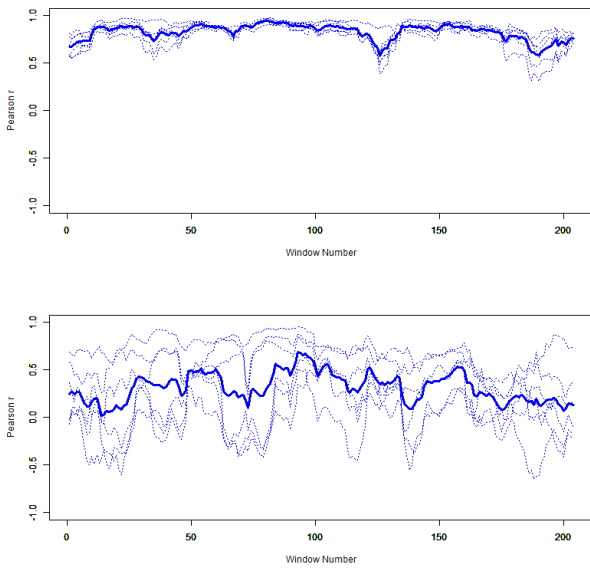


Figure 6: Graphiques des corrélations entre plusieurs régions du cerveau générés avec R pour deux sujets d'âges différents. Le sujet de 64 ans est en haut, celui de 88 ans est en bas et démontre une plus forte variation des corrélations.

Pratique usuelle

Nos collaborateurs neuroscientifiques cherchent à identifier des motifs dans les IRMf afin de mieux comprendre les déficits associés à la maladie de Parkinson. Les réseaux pondérés dynamiques qu'ils ont analysés avec LinkWave étaient ceux sur lesquels ils travaillent habituellement, extraits de données IRMf et "nettoyés" selon les procédures généralement recommandées par leur communauté de recherche. Les liens pondérés représentent dans ces réseaux les corrélations entre certaines régions d'intérêt (ROIs) qu'ils ont eux-même spécifié. Les outils de visualisation existants sont peu nombreux et présentent généralement le réseau dans le contexte de la structure anatomique du cerveau, représentée en 3D (Figure 2). Les croisements de liens rendent alors difficile l'identification de motifs.

La pratique d'analyse actuelle de nos collaborateurs neuroscientifiques est essentiellement fondée sur la formation d'hypothèses et leur acceptation ou réfutation sur la base de tests et modèles statistiques plus ou moins avancés. Durant leurs analyses, nos collaborateurs ont généré un certain nombre de graphiques simples utilisant les environnements tels que R (voir Figure 6 par exemple). Ils représentent dans une même vue toutes les corrélations entre ROIs (*i.e.* tous les liens pondérés) pour deux participants différents. Ces simples graphiques (non interactifs) leur ont déjà permis de découvrir des motifs intéressants telle qu'une plus forte variabilité des corrélations chez les sujets plus âgés. D'autres découvertes de ce genre clairement liées à la visualisation des données ont amené nos collaborateurs à former de nouvelles hypothèses et à effectuer des analyses qu'ils n'avaient pas envisagées avant. Cette prise de conscience du potentiel lié à la visualisation a par la suite joué un rôle moteur dans notre collaboration.

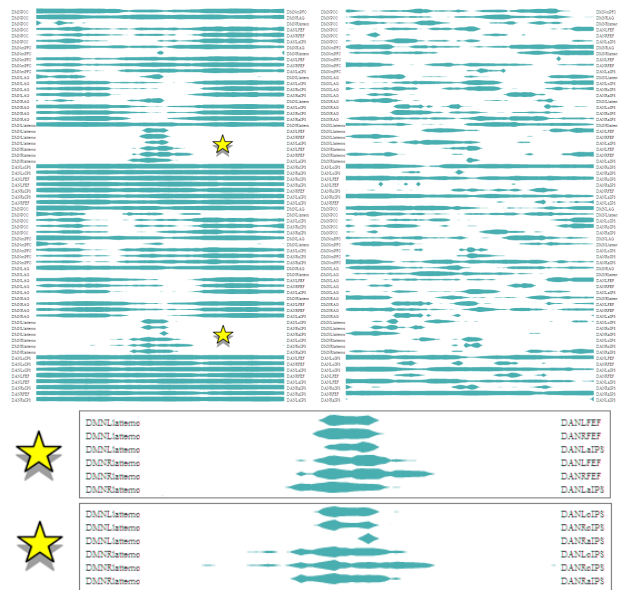


Figure 7: Représentations générées avec LinkWave pour les mêmes données que les graphiques de la Figure 6. Le sujet de 64 ans est à gauche, celui de 88 ans est à droite. La forte variation des poids des liens est identifiable par les variations des diagrammes en aires sur l'image de droite. L'étoile indique un nouveau motif découvert par nos collaborateurs. Il s'agit de liens issus de la région latérale temporelle (lattemp) dont l'évolution des poids diffèrent des autres.

Utilisation de LinkWave

Nous avons conçu et implémenté un prototype de LinkWave (Figure 4) puis raffiné les interactions proposées de manière itérative avec les neuroscientifiques sur un intervalle de trois mois. Durant ces trois mois, les 3 itérations majeures ont concerné la paramétrisation des calculs des fenêtres de corrélation (afin qu'ils puissent contrôler la taille de ces fenêtres), l'implémentation de plusieurs algorithmes de tri de la liste visuelle des liens, et l'ajout de vues synchronisées avec LinkWave (vue matricielle et vue noeuds-liens). Nous leur avons ensuite livré le logiciel qu'ils ont pu utiliser pendant les trois mois suivant pour analyser leurs données. Dans ce qui suit, nous décrivons les principaux commentaires et découvertes faits avec le système par les neuroscientifiques.

Confirmation de découvertes précédentes

Durant les premières utilisations de LinkWave, nos collaborateurs ont tout d'abord cherché à retrouver un certain nombre de découvertes réalisées auparavant. La première analyse fut faite pour mettre en avant la différence de synchronisation due à l'âge des sujets. Nos collaborateurs avaient en particulier généré avec R des graphiques mettant en évidence la variation des corrélations chez les sujets les plus âgés (Figure 6). Utilisant ces mêmes données dans LinkWave, nos collaborateurs ont créé les représentations présentées dans la Figure 7. La différence de variabilité des poids est directement reflété dans LinkWave mais ils ont également fait une seconde découverte. Les liens d'une des régions chez le sujet moins âgé présentent un motif différent des autres, ce qui n'était pas visible dans les graphiques précédents. Intrigués par cette région, ils explorèrent alors des données de participants moins âgés. A ce jour, ils continuent de poursuivre cette

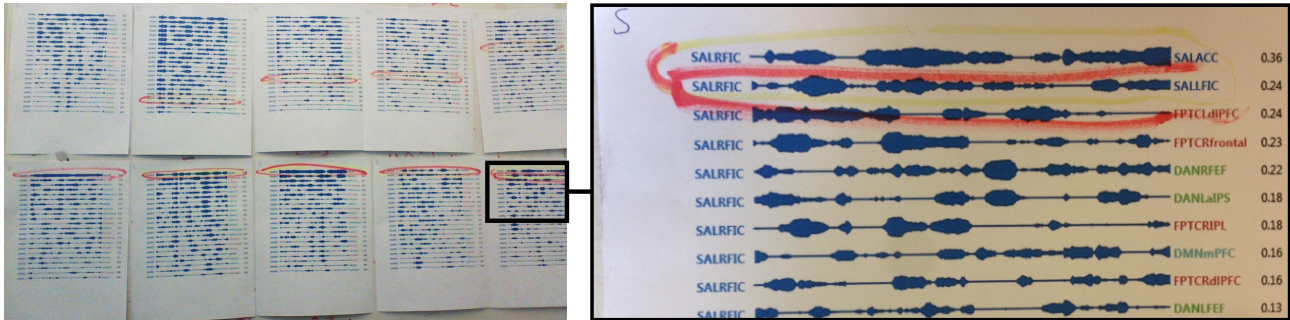


Figure 8: Impréssions annotées de la fenêtre principale de LinkWave. La rangée du haut présente un sous-ensemble de liens pour cinq participants du groupe contrôle ; la rangée du bas ce même sous-ensemble pour cinq participants du groupe de Parkinson. Les liens sont ordonnés par leur intensité moyennes, le lien entouré semble avoir un poids plus important dans le groupe de Parkinson.

direction de recherche. Cette nouvelle découverte leur a donc permis de faire de nouvelles hypothèses sur des données analysées auparavant.

Nos collaborateurs ont aussi cherché à reproduire la découverte d'une différence entre les sujets d'un groupe de contrôle et les sujets atteints de la maladie Parkinson. En réordonnant les liens par leur poids moyen et en comparant les sujets des deux groupes à l'aide de versions papier de visualisations LinkWave (Figure 8), ils ont identifié un lien spécifique entre deux ROIs qui s'avère beaucoup plus fort dans les sujets avec Parkinson que dans les sujets du groupe de contrôle. Ce lien est entouré sur les documents imprimés et apparaît en haut de la liste. Ce résultat n'était pas surprenant au vu de leur analyse initiale, mais il furent surpris qu'une interaction de quelques minutes mette en évidence ce motif découvert après plusieurs jours d'analyses statistiques. LinkWave leur permit en outre de continuer leur exploration et les mécanismes de filtrage par noeud source et destination leur permit d'inspecter systématiquement les liens de ces deux ROIs au reste du réseau, pour essayer d'identifier des motifs distinctifs entre le groupe contrôle et le groupe Parkinson.

Remise en question du seuillage des liens pondérés

Comme mentionné précédemment, il est commun de réduire le nombre de liens du réseau en filtrant les liens dont le poids est inférieur à une valeur donnée. Ce principe de seuillage est particulièrement utile pour les techniques de visualisation basées sur les diagrammes noeuds-liens car il peut améliorer leur lisibilité. Comme LinkWave ne bénéficie pas directement du filtrage des liens à un instant donné, nous n'avons pas implémenté ce seuillage. Cette décision a permis à nos collaborateurs de découvrir que, dans certains cas, des motifs intéressants émergent lorsque tous les poids sont conservés. La Figure 9 illustre ce type de découverte. Ces explorations avec LinkWave ont donc permis à nos collaborateurs de remettre en question cette pratique de seuillage et de considérer le recours à l'exploration visuelle en amont de l'analyse pour mieux comprendre l'impact du nettoyage des données.

Découverte de points de synchronisation

La découverte la plus importante faite par nos collaborateurs avec le système est l'identification de points temporels pour lesquels les poids de nombreux liens augmentent ou diminuent. Ces "points de synchronisation" se traduisent par des lignes verticales visibles à tous les niveaux

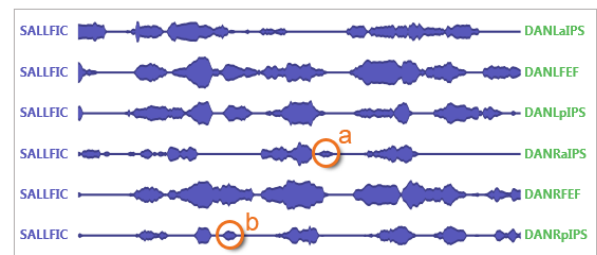


Figure 9: Filtrer les poids faibles peut permettre de réduire le bruit mais peut aussi cacher certains motifs. Dans cet exemple le motif annoté (a) est certainement du bruit. Par contre ; le motif annoté (b) peut se révéler être un motif car il fait écho à un motif dans les deuxième, troisième et cinquième lignes.

d'agrégation et illustrés par des flèches sur la Figure 10. Bien que ce type de synchronisation soit une hypothèse discutée par les neuroscientifiques depuis plusieurs années, ils étaient extrêmement surpris et enthousiastes lors de cette découverte, demandant aux autres chercheurs et étudiants présents dans le laboratoire de venir voir les visualisations sur leurs écrans.

Au cours des discussions qui suivirent leur réaction fut de vérifier que ces motifs ne soient pas dus à un bogue du prototype ou à un artefact durant l'acquisition des données. Après plusieurs analyses complémentaires, ils ne furent pas en mesure de corréliser ces motifs avec les informations physiologiques collectées lors de l'acquisition (mouvements du sujet, respirations, etc.). A ce jour, nos collaborateurs continuent leurs analyses de ce phénomène de manière plus systématique pour comprendre ce que représentent ces points de synchronisation et leur importance dans le fonctionnement du cerveau.

Découverte de séquençement

Un autre type de motif qui intéresse nos collaborateurs est illustré dans la Figure 10. Il s'agit de séquençements possibles dans les forces de connexions entre deux groupes de liens. Ces séquençements peuvent indiquer un rôle particulier ou une synchronisation entre des régions d'intérêt. Ces motifs, difficiles à identifier dans la représentation individuelle des liens, deviennent apparents après agrégation noeud-groupe.

En résumé, l'utilisation du prototype LinkWave a permis à

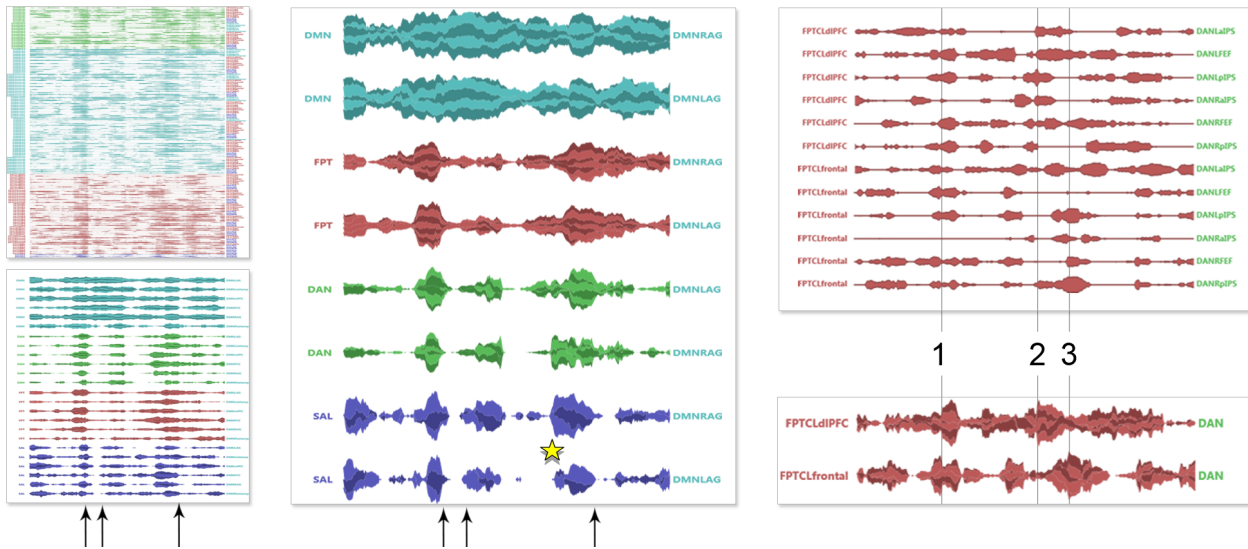


Figure 10: Les points de synchronisation sont annotés par des flèches dans ces figures. Il s'agit d'un instant donné auquel les poids des liens suivent une forte variation. L'étoile sur l'image du milieu indique une synchronisation plus subtile où le poids du lien placé à l'avant-dernière ligne semble augmenter légèrement avant les autres. Ce type de séquençement est difficile à capturer dans une vue non agrégée des liens. L'image de droite est un autre exemple qui indique ces motifs de séquençement et possible synchronisation entre certaines régions du cerveau. Les liens concernent la connectivité de deux régions FPTCLdLPFS et FPTCLfrontal au groupe DAN. Au temps 1, la force de connectivité des deux régions au groupe DAN s'inverse soudainement : FPTCLfrontal est connecté moins fortement alors que FPTCLdLPFS devient plus fortement connecté. Un autre de ces changements est plus progressif entre les temps 2 et 3.

des neuroscientifiques de découvrir des motifs temporels dans leurs données et surtout de former des hypothèses et d'effectuer des analyses qu'ils n'avaient pas envisagées auparavant. Il est intéressant de noter que les vues synchronisées, sollicitées par nos collaborateurs ne semblent pas avoir été utilisées lors des découvertes. D'après leurs commentaires, elles leur ont cependant permis de se familiariser avec la représentation visuelle de LinkWave pendant les premiers jours d'utilisations du système et de les aider à interpréter les nouveaux motifs lors de l'exploration. Bien que de nombreuses recherches et analyses systématiques soient encore nécessaires pour conclure sur la validité et l'importance des motifs découverts avec notre outil, ce premier retour d'expérience a permis de démontrer le potentiel de ce nouveau paradigme d'exploration visuelle des réseaux pondérés dynamiques.

TRAVAUX ANTÉRIEURS

Nous décrivons les travaux antérieurs dans le domaine de la visualisation d'information et non spécifiques au domaine des neurosciences.

Représentations de réseaux dynamiques

Les réseaux ont deux représentations visuelles communes : les diagrammes nœuds-liens [5] et les matrices d'adjacence [21]. L'animation est une technique standard pour adapter ces représentations aux graphes dynamiques [6]. Mais bien que l'utilisation de la dimension temporelle paraîsse naturelle pour représenter des données dynamiques, l'efficacité de ces animations a été remise en cause [4, 16, 17]. En particulier, Tversky *et al.* ont démontré que l'identification des changements entre plusieurs images successives est laborieuse étant donné que le cerveau humain ne peut suivre qu'un petit nombre (cinq) d'objets en mouvement à la fois [31]. Les "small

multiples" [30], qui consistent à offrir une série de représentations pour chaque point dans le temps, sont une alternative à l'animation. Cette technique permet d'accomplir certaines tâches plus efficacement, comme la comparaison de topologies à des instants éloignés [4]. Mais elle ne passe pas à l'échelle lorsque le nombre d'instants considérés augmente. Quelques travaux ont proposé de représenter le temps comme une troisième dimension [2, 8], mais les techniques proposées souffrent des habituels problèmes d'occultation inhérents aux représentations 3D.

Plusieurs chercheurs se sont intéressés au cas particulier des réseaux pondérés dynamiques. Burch *et al.* [12] proposent une technique basée sur les coordonnées parallèles [22]. Leur technique représente les nœuds du réseau par des lignes verticales répliquées pour chaque point considéré dans le temps. Les liens pondérés connectent deux lignes verticales, indiquant le poids par l'intensité du lien. Malheureusement, la lisibilité de la représentation dépend fortement de la densité du réseau et la technique proposée ne permet pas aisément de caractériser l'évolution du poids des liens. Les autres techniques proposées sont basées sur les matrices. Perer and Sun proposent une approche basée sur les "small multiples" [25]. Yi *et al.* [34] et Brandes & Nick [9] définissent des glyphes pour représenter l'évolution des poids des liens. Les glyphes sont une représentation visuelle (telle qu'un histogramme) de petite taille qui peut donc être insérée dans la cellule correspondante de la matrice. L'utilisation de glyphe permet d'ajouter une dimension de données aux matrices, mais rend difficile l'extraction de leurs motifs temporels car les glyphes ne sont pas alignés.

Le système le plus proche de nos travaux est Cubix [24]. Ce système est basé sur une matrice d'adjacence dont la

troisième dimension est utilisée pour le temps. Ce cube espace-temps permet aux utilisateurs de réaliser des opérations très avancées. Il est par exemple possible de créer une matrice d'adjacence standard mais de remplacer une colonne par son évolution dans le temps. Par contraste, LinkWave opte pour une des structures les plus simples : une liste réordonnable. Un avantage majeur de cette liste est la possibilité d'obtenir une vue d'ensemble d'un (très) grand nombre de liens en utilisant des interactions familières aux utilisateurs tels que pan et zoom, filtrage et tri. Notre système a été exclusivement conçu pour accomplir les tâches basées sur les liens du réseaux. Intégrer des fonctionnalités de tri plus avancées basées sur les différentes configurations de cubix, tout en préservant la simplicité de LinkWave est une direction intéressante.

Représentations de données temporelles

La représentation visuelle de données temporelles a donné lieu à une quantité considérable de travaux [3, 1]. Les techniques les plus communes pour représenter ce type de données sont les graphiques en aires ou d'horizons [20], ou les animations. Les représentations utilisées dans notre travail sont inspirées des graphiques empilés [32] et *streamgraphs* tels que *ThemeRiver* [19] ou *TouchWave* [7]. Plusieurs travaux antérieurs se sont concentrés sur l'identification de séquences et de motifs dans les données temporelles tels que *PatternFinder* [15] et *LifeLines* [26]. Ces systèmes permettent de créer et raffiner interactivement des requêtes afin de trouver des séquences particulières dans un grand nombre de données temporelles. Aucun de ces systèmes n'a été appliqué aux réseaux pondérés dynamiques.

Concernant l'exploration de structures plus complexes, plusieurs techniques se sont focalisées sur la représentation de données temporelles associées à des hiérarchies. *Timeline trees* [10], son équivalent circulaire *TimeRadar-Trees* [11] et *TimeEdgeTrees* [13] sont de bons exemples. Dans ces représentations, les relations entre les données temporelles sont visualisées par des diagrammes nœuds-liens. Les analystes peuvent ouvrir et fermer les branches de l'arbre interactivement, ce qui permet d'agréger ou de désagréger les visualisations temporelles. Ces techniques sont efficaces pour les structures hiérarchiques dans lesquelles les liens ne se croisent pas, mais ne sont pas appropriées aux réseaux plus complexes. Quelques travaux se sont enfin intéressés à représenter des données temporelles associées aux nœuds des réseaux [27], mais ces techniques ne permettent pas de représenter la temporalité des liens, de fait beaucoup plus nombreux.

CONCLUSION

Cet article présente les premiers résultats de notre collaboration avec des neuroscientifiques qui cherchent à identifier des motifs dans les réseaux pondérés dynamiques extraits de données d'imagerie par résonance magnétique fonctionnelle (IRMf). À l'issue d'entretiens et de séances d'observation, nous avons dressé une liste de tâches à accomplir pour permettre l'exploration des réseaux considérés. Ces tâches, centrées sur l'évolution des poids des liens, se retrouvent également dans d'autres domaines d'application. Comme les techniques de visualisation de réseaux actuelles ne sont pas centrées sur l'évolution

des poids mais plutôt l'évolution des nœuds, nous avons conçu un nouveau système, LinkWave, permettant d'accomplir ce type de tâches. LinkWave met en œuvre un nouveau paradigme de représentation des réseaux pondérés dynamiques dans lequel les liens et l'évolution de leur poids sont mis en avant. Les utilisateurs interagissent directement avec les liens et peuvent les agréger pour explorer la connectivité du réseau au cours du temps, et ce à plusieurs niveaux d'abstraction. Nous avons développé LinkWave en collaboration avec des neuroscientifiques et présenté dans cet article leurs retours d'expérience ainsi que quelques découvertes préliminaires.

Dans le cadre de notre collaboration avec ces neuroscientifiques, nous développons une deuxième version de LinkWave incluant de nouveaux mécanismes pour former des requêtes sur la topologie du réseau et sur les séquences temporelles. Nous explorons également des moyens de mieux intégrer les outils statistiques à notre système de visualisation. En effet, même si les motifs découverts par exploration visuelle permettent aux scientifiques de former de nouvelles hypothèses, ceux-ci ont toujours recours à leurs outils statistiques traditionnels pour les rejeter ou les valider. Une intégration de ces outils pourrait permettre un processus d'exploration plus efficace. Enfin, la direction de recherche la plus prometteuse est la conception d'interactions destinées à faciliter la comparaison de plusieurs liens ; groupes de liens ou réseaux. Cette tâche est particulièrement importante pour nos collaborateurs car ils explorent les données issues de plusieurs sujets et cherchent à identifier des motifs représentatifs.

REMERCIEMENTS

Nous remercions Sohaib Ghani qui a contribué au développement du système durant son stage à Microsoft Research. Nous remercions également Bongshin Lee qui a participé aux discussions sur la conception du système. This research was supported by grants from the National Institutes of Health 1RC4NS073008-01.

BIBLIOGRAPHIE

1. Adjanor, K., Lecolinet, E., Guiard, Y., and Ribière, M. Visualisation interactive de données temporelles: Un aperçu de l'état de l'art. In *Actes d'IHM '10*, ACM (2010), 97–104.
2. Ahmed, A., Dwyer, T., Murray, C., Song, L., and Wu, Y. X. WilmaScope graph visualisation. In *Proceedings of InfoVis '04*, IEEE (Oct. 2004), r4.
3. Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. Visualizing time-oriented data – a systematic view. *Computers and Graphics* 31, 3 (June 2007), 401–409.
4. Archambault, D., Purchase, H., and Pinaud, B. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE TVCG* 17, 4 (April 2011), 539–552.
5. Auber, D., Chiricota, Y., Delest, M., Melançon, G., Domenger, J.-P., and Mary, P. Visualisation de graphes avec Tulip : exploration interactive de grandes masses de données en appui à la fouille de données et à l'extraction de connaissances. In *Actes d'EGC '07*, Cépaduès (Jan. 2007), 147–156.

6. Bach, B., Pietriga, E., and Fekete, J.-D. GraphDiaries: animated transitions and temporal navigation for dynamic networks. *IEEE TVCG 20*, 5 (May 2014), 740–754.
7. Baur, D., Lee, B., and Carpendale, S. TouchWave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of ITS '12*, ACM (2012), 255–264.
8. Brandes, U., and Corman, S. R. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization 2*, 1 (Mar. 2003), 40–50.
9. Brandes, U., and Nick, B. Asymmetric relations in longitudinal social networks. *IEEE TVCG 17*, 12 (Dec. 2011), 2283–2290.
10. Burch, M., Beck, F., and Diehl, S. Timeline trees: Visualizing sequences of transactions in information hierarchies. In *Proceedings of AVI '08*, ACM (2008), 75–82.
11. Burch, M., and Diehl, S. TimeRadarTrees: visualizing dynamic compound digraphs. In *Proceedings of EuroVis '08*, Eurographics Association (2008), 823–830.
12. Burch, M., Vehlow, C., Beck, F., Diehl, S., and Weiskopf, D. Parallel edge splatting for scalable dynamic graph visualization. *IEEE TVCG 17*, 12 (Dec 2011), 2344–2353.
13. Burch, M., and Weiskopf, D. Visualizing dynamic quantitative data in hierarchies - TimeEdgeTrees: attaching dynamic weights to tree edges. In *IMAGAPP/IVAPP*, SciTePress (2011), 177–186.
14. Ellis, W. D. *A source book of Gestalt psychology*. Routledge & Kegan Paul, 1938.
15. Fails, J., Karlson, A., Shahamat, L., and Shneiderman, B. A visual interface for multivariate temporal data: finding patterns of events across multiple histories. In *Proceedings of VAST '06*, IEEE (2006), 167–174.
16. Farrugia, M., and Quigley, A. Effective temporal graph layout: A comparative study of animation versus static display methods. *Information Visualization 10*, 1 (Jan. 2011), 47–64.
17. Ghani, S., Elmqvist, N., and Yi, J. S. Perception of animated node-link diagrams for dynamic graphs. In *Proceedings of EuroVis '12*, John Wiley & Sons (June 2012), 1205–1214.
18. Ghoniem, M., Fekete, J.-D., and Castagliola, P. Comparaison de la lisibilité des graphes en représentation noeuds-liens et matricielle. In *Actes d'IHM 04*, ACM (2004), 77–84.
19. Havre, S., Hetzler, B., and Nowell, L. ThemeRiver: visualizing theme changes over time. In *Proceedings of InfoVis '00*, IEEE (Oct. 2000), 115–123.
20. Heer, J., Kong, N., and Agrawala, M. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of CHI '09*, ACM (2009), 1303–1312.
21. Henry, N., and Fekete, J.-D. MatrixExplorer: un système pour l'analyse exploratoire de réseaux sociaux. In *Actes d'IHM '06*, ACM (2006), 67–74.
22. Inselberg, A. The plane with parallel coordinates. *The Visual Computer 1*, 2 (1985), 69–91.
23. Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D., and Henry, N. Task taxonomy for graph visualization. In *Proceedings of BELIV '06*, ACM (2006), 1–5.
24. Melo, C., Mikheev, A., Le-Grand, B., and Aufaure, M.-A. Cubix: A visual analytics tool for conceptual and semantic data. In *Proceedings of ICDMW '12*, IEEE (Dec. 2012), 894–897.
25. Perer, A., and Sun, J. MatrixFlow: temporal network visual analytics to track symptom evolution during disease progression. In *Proceedings of AMIA '12*, American Medical Informatics Association (Nov. 2012), 716–725.
26. Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. Lifelines: Visualizing personal histories. In *Proceedings of CHI '96*, ACM (1996), 221–227.
27. Saraiya, P., Lee, P., and North, C. Visualization of graphs with associated timeseries data. In *Proceedings of InfoVis '05*, IEEE (Oct. 2005), 225–232.
28. Shneiderman, B., and Plaisant, C. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, ACM (New York, NY, USA, 2006), 1–7.
29. Sporns, O. *Networks of the Brain*. MIT Press, Oct. 2010.
30. Tufte, E. *Envisioning Information*. Graphics Press, May 1990.
31. Tversky, B., Morrison, J. B., and Betrancourt, M. Animation: can it facilitate? *IJHCS 57*, 4 (Oct. 2002), 247–262.
32. Wattenberg, M. Baby names, visualization, and social data analysis. In *Proceedings of InfoVis '05*, IEEE (Oct. 2005), 1–7.
33. Xia, M., Wang, J., and He, Y. BrainNet Viewer: a network visualization tool for human brain connectomics. *PLoS ONE 8*, 7 (07 2013), e68910.
34. Yi, J. S., Elmqvist, N., and Lee, S. TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations. *IJHCI 26*, 11 (Nov. 2010), 1031–1051.

PLACID : un planificateur pour composer dynamiquement des Services IHM

Yoann Gabillon

CRP - Gabriel Lippmann
41, Rue du Brill
L-4422 Belvaux, Luxembourg
gabillon@lippmann.lu

Gaelle Calvary

Univ. Grenoble Alpes, LIG
F-38000 Grenoble, France
CNRS, LIG, F-38000
Grenoble, France
gaelle.calvary@imag.fr

Humbert Fiorino

Univ. Grenoble Alpes, LIG
F-38000 Grenoble, France
CNRS, LIG, F-38000
Grenoble, France
humbert.fiorino@imag.fr

RÉSUMÉ

La Composition Dynamique de Services a pour but de composer un système interactif à partir d'un ensemble de services disponibles correspondant à des composants. Un composant est constitué d'une partie fonctionnelle et d'une partie Interface Homme-Machine (IHM). En Génie Logiciel, la grande majorité de la littérature se concentre sur la composition dynamique de services fonctionnels. Si l'on fait l'hypothèse qu'un service IHM peut aussi être composé, cela entraîne une nouvelle problématique de recherche en Interaction Homme-Machine : la Composition Dynamique de Services IHM. Cet article présente un algorithme de planification permettant de résoudre le problème de la composition dynamique d'IHM pour produire le modèle de tâches de l'IHM composée permettant à l'utilisateur d'atteindre son objectif.

ACM Classification Keywords

H.5.2 User Interfaces: Ergonomics, Graphical user interfaces (GUI), Prototyping, User-centered design.

General Terms

Design, Human factors, Algorithms

Mots Clés

Composition Dynamique, Interface Homme-Machine (IHM), service IHM, composition d'IHM, Formalisation, Algorithme

INTRODUCTION

La Composition Dynamique de Services (DSC pour Dynamic Service Composition) a pour objectif de composer un système interactif à partir d'un ensemble de services disponibles et d'un objectif utilisateur à satisfaire. Le processus DSC peut être décomposé en deux étapes. Premièrement, le plan (i.e. une sélection et un ordonnancement d'un ensemble de services) est calculé. Deuxièmement, les composants sont composés conformément au plan calculé pour produire un système interactif exécutable.

Un système interactif comprend un noyau fonctionnel (FC pour Functional Core) et une partie Interface Homme-Machine (IHM) [5]. De la même manière, une distinction peut être effectuée entre service fonctionnel et IHM [20]. Les services fonctionnels sont fournis par des composants fonctionnels. Les services IHM sont fournis par les composants IHM. Si l'on fait l'hypothèse que les services IHM peuvent être composés comme les services fonctionnels, cela entraîne une nouvelle problématique de recherche en Interaction Homme-Machine : la Composition Dynamique de Services IHM.

En Génie Logiciel (GL), la majeure partie des travaux s'intéresse à la composition dynamique de services fonctionnels. Le reste du temps, les services IHM sont soit considérés comme des services fonctionnels, soit l'IHM est composée dans un deuxième temps. Cependant, pour promouvoir une composition dynamique centrée utilisateur, les services IHM doivent aussi être composés dynamiquement pour produire un plan d'exécution.

L'article présente comme contribution principale la formalisation et l'implémentation d'un algorithme de planification pour produire un plan de service IHM sous la forme d'un arbre de tâches. En effet, la composition dynamique nécessite une technique d'Intelligence Artificielle (IA) pour sélectionner et ordonner les services à composer. En conséquence, nous avons utilisé les concepts de planification automatique déjà utilisées en DSC pour produire un plan sous la forme d'un modèle de tâches. L'algorithme PLACID (pour « PLAnner for dynamically Composing user Interfaces Descriptions ») permet de calculer un plan de services IHM. PLACID est intégré dans le prototype COMPOSE qui compose dynamiquement une IHM à partir de ce plan.

ÉTAT DE L'ART ET POSITIONNEMENT

La Composition de Services peut être *statique* ou *dynamique* [3]. La composition statique débute à partir d'un plan (un schéma de composition de services) pour produire un système composé à partir de composants préexistants alors que la composition dynamique a pour objectif de calculer ce plan à partir d'un objectif utilisateur et de services disponibles [3]. La DSC repose sur l'hypothèse que chaque service peut être vu comme une action décrite par ses préconditions, postconditions, et effets [22]. Ces actions sont utilisées pour calculer un plan selon le contexte d'usage courant (propriétés concernant l'utilisateur, sa plate-forme et son environnement [24]).

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670448>

Cette section propose une vue d'ensemble des techniques d'IA utilisés en GL et en IHM pour composer dynamiquement des services fonctionnels et IHM.

Génie Logiciel

La Composition Dynamique de Services (web services) nécessite une technique de l'IA pour calculer un plan. Par exemple, la Planification Automatique [9] développe des algorithmes pour atteindre un objectif à partir d'actions disponibles. En GL, différentes approches (techniques d'IA) ont été explorées pour calculer un plan de services comme le Calcul de Situations, la Planification à base de Règles, la Preuve de Théorème ou la Planification Automatique [22].

Calcul de Situations. Par exemple, [14] adopte le langage Golog pour calculer automatiquement un plan qui est une séquence d'actions. Golog est un langage de programmation qui représente les changements ou les évolutions de situations, actions ou objets. L'objectif utilisateur est exprimé en prédicats de la logique de premier ordre. Les services sont des actions. À partir de règles logiques et de contraintes, le plan est calculé à l'exécution.

Planification à base de Règles. Par exemple, [15] calcule un service composite à partir de règles de composabilité pour vérifier si les services peuvent être composés ou non. Premièrement, à partir d'une description de haut niveau, l'ensemble des compositions possibles est décrit. Deuxièmement, à partir des règles de composition, les descriptions de haut niveau sont composées. Troisièmement, les services sont sélectionnés à partir des descriptions pour obtenir un plan.

Preuve de Théorème. Par exemple, [21] utilise un prouveur de théorèmes. Les services disponibles et l'objectif utilisateur sont traduits en prédicats du premier ordre. Puis le plan est calculé par le prouveur de théorèmes qui utilise ces prédicats.

Planificateur PDDL. Classiquement, un planificateur PDDL manipule trois entrées : un but, (ex : « J'ai besoin d'aller de mon hôtel au bureau du médecin. »), l'état initial du monde (ex : Victor est à Philadelphie) et un ensemble d'actions (ex : voyager en voiture). Une action est définie par des préconditions (un ensemble de prédicats devant être vrais) et les effets sur l'état du monde (ex : pour l'action « travel by car », les effets seront que Victor sera dans le bureau du médecin). En retour, les algorithmes de planification automatique calculent un plan (i.e., une séquence d'actions) pour atteindre le but. Par exemple, [12] utilise un planificateur PDDL pour produire un plan de services permettant à l'utilisateur d'atteindre son objectif.

Planificateur HTN. Hierarchical Task Network (HTN) [6] est une approche hiérarchique de planification automatique. En HTN, les dépendances entre actions peuvent être données sous la forme d'un réseau. En conséquence, HTN permet l'utilisation de descriptions hiérarchiques pour décrire les actions abstraites (appelée méthodes) en sous-actions pour obtenir des actions élémentaires (i.e., correspondant aux actions PDDL). Une méthode a des préconditions et un ensemble de sous-actions. Par

exemple, la méthode « voyager » peut être décomposée en deux sous-actions « voyager en voiture » ou « voyager en avion ». À partir d'un état initial, un planificateur HTN essaye d'appliquer les méthodes et les actions jusqu'à ce que l'objectif utilisateur soit réalisable par une séquence d'actions. Par exemple, [23] utilise un planificateur HTN comme JShop [6] pour produire un plan à partir d'une hiérarchie d'actions disponibles.

De même, le projet européen ATRACO (Adaptive and Trusted Ambient Ecologies) [1] utilise un planificateur HTN pour permettre de décrire une hiérarchie de tâches pour différencier les tâches pouvant être exécutées en parallèle ou en séquence. En effet, ce planificateur utilise aussi des concepts de Partial-Order Planning pour produire un plan d'actions concrètes partiellement ordonnées. Ainsi, il est possible de composer des services IHM en parallèle, par contre, seules les actions apparaissent dans le plan (les méthodes sont omises).

Pour produire une IHM, la composition de composants IHM peut également être effectuée après la DSC. L'idée est de tirer avantage du plan calculé dynamiquement pour composer, dans un second temps, les IHM des services. Par exemple, [7] propose de composer les IHM des services sélectionnés dans une même fenêtre. De cette manière, l'IHM composée se déduit du plan de services fonctionnels.

CRUSE [20] introduit la notion de services IHM. Il s'intéresse à la composition de composants IHM à partir de services IHM en proposant une architecture système. Cependant, CRUSE ne se concentre pas sur la sélection des services IHM (i.e. le calcul du plan). En effet, il suppose le plan déjà calculé.

Interaction Homme-Machine

À haut niveau d'abstraction, une IHM est modélisée par un *modèle de tâches* [18, 4]. Le modèle de tâches (TM pour Task Model) est une description réursive de la *tâche utilisateur* en sous-tâches parallèles ou séquentielles jusqu'à obtenir des *tâches élémentaires* (i.e., des tâches qui ne peuvent être décomposées qu'en actions physiques). Par exemple, pour obtenir une assistance médicale « get medical assistance », l'utilisateur doit appeler le médecin « call the office » et suivre le trajet pour s'y rendre « find route information ». Une tâche peut être vue comme un *service IHM* fourni par un composant IHM. En conséquence, pour composer dynamiquement un nouveau service IHM, le plan calculé doit être un arbre de tâches au lieu de seulement une séquence. La correspondance entre les services et les actions est effectuée à l'aide d'ontologies [10].

Une IHM adaptative est une IHM capable de s'adapter ou d'être adaptée au contexte d'usage courant. Beaucoup de travaux se concentrent sur l'adaptation d'une IHM à l'exécution [4, 16]. Cependant, tous ces travaux démarrent à partir d'un modèle de tâches complet. Or le modèle de tâches, i.e., le plan en termes de services IHM, doit être calculé à l'exécution.

Huddle [17] utilise un planificateur PDDL pour produire une séquence d'actions qui représente une configuration

valide des composants IHM à composer. L'objectif est de générer automatiquement une IHM. Cependant, le planificateur ne manipule pas d'actions abstraites à l'instar d'un planificateur HTN.

Paterno et al. [19] montrent comment produire dynamiquement une IHM à partir de services Web. En particulier, les auteurs présentent comment créer une description abstraite d'une IHM à partir du modèle de tâches lié avec des services Web existants. L'objectif est de gérer l'implémentation de l'IHM. De la même manière, Kritikos et al. [13] n'utilisent pas un planificateur mais supposent un modèle de tâches complet pour produire un plan qui décrit comment les services existants peuvent être composés pour générer l'IHM. En conséquence, ces deux travaux [19, 13] ne se concentrent pas sur l'algorithme pour calculer le plan.

La composition par composants IHM se base sur l'hypothèse que le schéma de composition (le plan en termes de services) est connu et propose des frameworks ou opérateurs pour aider le concepteur à produire une IHM composée. Par exemple, [11] propose de composer des composants IHM conformément au schéma de composition fonctionnel et propose une formalisation de cette problématique. De manière analogue, [2] compose des composants IHM décrits et annotés à l'aide d'ontologies.

Positionnement

La contribution de cet article porte sur un algorithme permettant de calculer un plan de services IHM. Les IHM étant modélisées à plus haut niveau d'abstraction en tâches utilisateur, le plan qui décrit une IHM composée doit être un modèle de tâches comportant des tâches en parallèle et en séquence. Les travaux portant sur la composition dynamique produisent un plan sous la forme d'une séquence d'actions. A notre connaissance, seul le planificateur du projet ATRACO[1] produit un plan partiellement ordonné. Cependant, ce plan ne comporte pas les méthodes (actions abstraites) fournies en entrée. En conséquence, nous proposons de manière originale un algorithme de planification permettant de produire un plan sous la forme d'un modèle de tâches.

FORMALISATION DU PROBLÈME ET DE LA SOLUTION

Formalisation du problème

Le principe est de produire une IHM dont la fonction est de faire passer l'utilisateur de sa situation courante à une situation dans laquelle son objectif est satisfait. Plus formellement, la situation courante (le contexte d'usage courant) est l'état initial s_0 d'un graphe d'état. Un état est représenté par un ensemble de propositions logiques. Ici, $s_0 = \{has(Victor, Smartphone), internet(Smartphone), at(Victor, Philadelphia), Guidance_Prompting(), \dots\}$

De la même manière, l'objectif de l'utilisateur est représenté par un ensemble de propositions g . L'état contenant g est appelé l'état but s_g : $s_g = \{found(Victor, medical_assistance), \dots\}$.

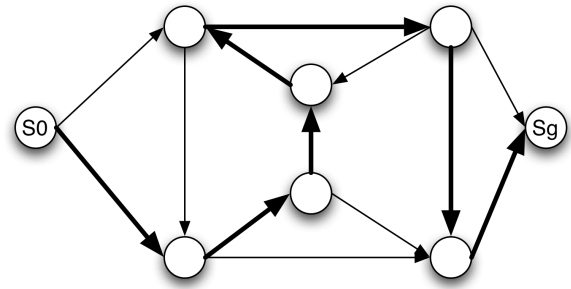


Figure 1. Graphe des changements d'état.

Le graphe d'état est un graphe des changements d'état dont les nœuds sont les états et les arcs sont les actions qui permettent à l'utilisateur de changer d'état à l'aide d'une IHM (figure 1).

Une action a est définie par le triplet $(precond(a), add(a), del(a))$ où $precond(a)$ est un ensemble de propositions logiques qui représente les conditions d'application de l'action : une action est applicable si et seulement si l'état courant contient $precond(a)$. Nous notons $precond^+(a)$ et $precond^-(a)$ respectivement les propositions logiques positives et négatives. $add(a)$ et $del(a)$ sont les effets respectivement ajoutés et supprimés à l'état courant. Si une action a est applicable, l'état suivant peut être calculé par :

$$\gamma(s_i, a) = (s_i - del(a)) \cup add(a) = s_{i+1}$$

En conséquence, le problème est de construire une IHM qui permet d'atteindre l'état but à partir de l'état initial. Soit π une liste d'actions de longueur k et s l'état courant. L'état suivant est défini par :

$$\gamma(s, \pi) = \begin{cases} s, & \text{si } k = 0 \\ \gamma(\gamma(s, a_1), [a_2, \dots, a_k]), & \text{si } k > 0 \text{ et} \\ & a_1 \text{ est applicable} \\ & \text{à } s \end{cases}$$

En pratique, l'utilisateur exprime ses besoins/son objectif. Ils sont traduits en un ensemble de propositions logiques g . Le système perçoit le contexte d'usage courant et le représente sous la forme d'un ensemble de propositions logiques s_0 . Par exemple, le fait qu'un utilisateur nommé Victor possède un mur numérique est exprimé par la formule logique $has(Victor, Smartphone)$. De la même manière, l'ensemble des informations caractérisant le contexte d'usage courant sont traduites en propositions logiques. L'algorithme de planification connaît les actions que l'utilisateur peut accomplir grâce aux IHM disponibles. Les actions sont définies en « intention » sous la forme d'opérateurs et de méthodes.

Un opérateur o est de la forme $(precond(o), add(o), del(o))$. Il est défini par ses préconditions qui sont un ensemble de prédicats en logique du premier ordre. $add(a)$ et $del(a)$ sont respectivement les effets ajoutés (supprimés) à l'état courant. Par exemple, les préconditions de l'opérateur $Call_the_office(?u, ?p)$ peuvent être les prédicats « $has(?u, ?p)$ » et « $internet(?p)$ » ; l'utilisateur $?u$ et

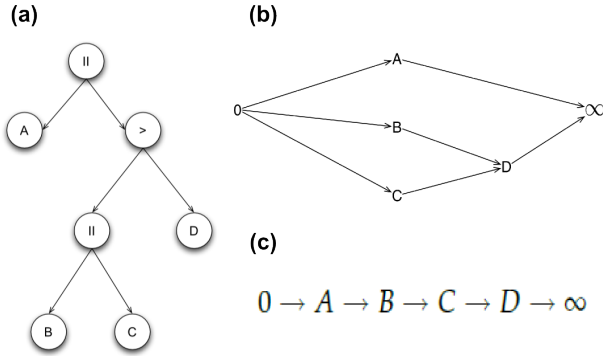


Figure 2. (a) Modèle de tâches calculé par le planificateur ; (b) La projection ρ du modèle de tâches ; (c) linéarisation.

la plate-forme $?p$ sont les variables du domaine. Donc, l'opérateur $\ll Call_the_office(?u, ?p) \gg$ peut représenter les actions $\ll Call_the_office(Victor, Smartphone) \gg$, $\ll Call_the_office(Victor, Wall) \gg$, etc.

Une méthode m est de la forme $(precond(m), type(m), subAct(m))$ où $precond(m)$ définit ses préconditions et $subAct(m)$ un ensemble d'opérateurs ou de méthodes à accomplir pour réaliser m . Par exemple, une méthode $\ll se\ déplacer\ en\ taxi \gg$ peut se décomposer en trois opérateurs $\ll appeler\ le\ taxi \gg$, $\ll prendre\ le\ taxi \gg$, et $\ll payer\ le\ taxi \gg$. Le type de la décomposition est spécifié dans $type(m)$ qui est soit une séquence, soit un parallélisme. En conséquence, les opérateurs ou méthodes de $subAct(m)$ sont totalement ordonnés. S'ils sont en parallèle, les éléments de $subAct(m)$ peuvent être réalisés peu importe l'ordre. Par exemple, la méthode $\ll Get_medical_assistance \gg$ se décompose en parallèle par $\ll Call_the_office \gg$ et $\ll Find_route_information \gg$.

Formalisation de la solution

À partir de l'état initial s_0 , de l'objectif utilisateur g et d'un ensemble de méthodes et d'opérateurs, l'algorithme de planification calcule un modèle de tâches qui permet à l'utilisateur d'atteindre son objectif. Ce modèle de tâches est un arbre. Chaque nœud correspond à un composant IHM existant. Par exemple, la tâche $\ll Call_the_office \gg$ correspond à un composant IHM permettant d'appeler quelqu'un. Une fois calculé, le modèle de tâches permet de savoir les composants IHM à composer et comment. Par exemple, le composant $\ll Get_medical_assistance \gg$ contiendra les composants $\ll Call_the_office \gg$ et $\ll Find_route_information \gg$.

Le modèle de tâches contient deux types de nœuds : interne ou feuille. Un nœud interne est une méthode complètement instanciée (toutes les variables sont liées avec une constante) de type séquence ($\ll \gg$) ou parallélisme ($\ll II \gg$). Une feuille est une action, i.e. un opérateur totalement instancié (figure 2a). Nous appelons projection ρ du modèle de tâches un ensemble partiellement ordonné d'actions (figure 2b). La relation d'ordonnancement $\ll C \gg D$ exprime que l'action C précède l'action D . 0 et ∞ sont les actions formelles qui représentent le début et la fin de l'interaction. La linéarisation de ρ est un ensemble composé des

mêmes actions complètement ordonnées par la relation de précédence. Un exemple de linéarisation est présenté dans la figure 2c. $L(\rho)$ est l'ensemble de linéarisations de ρ . Le problème de la composition est formalisé par : $P = (s_0, g, A)$ où s_0 est l'état initial, g l'objectif et A un ensemble d'actions possibles (en pratique exprimé sous la forme d'opérateurs et de méthodes). Résoudre ce problème revient à trouver une solution ρ (en pratique un modèle de tâches) tel que : $\forall \pi \in L(\rho), g \subseteq \gamma(s_0, \pi)$

Intuitivement, pour trouver une solution, il est nécessaire de calculer l'ensemble des linéarisations (toutes les séquences d'actions) qui permettent à l'utilisateur d'atteindre son objectif. En effet, une linéarisation correspond à une utilisation possible des actions (tâches) de l'IHM composée. Pour qu'une IHM composée soit solution, il faut qu'elle permette à l'utilisateur de réaliser la séquence (linéarisation) qu'il souhaite. Par exemple, deux actions $\ll a1 = Call_the_office \gg$ et $\ll a2 = Find_route_information \gg$ peuvent être exécutées en parallèle si l'utilisateur peut exécuter les deux linéarisations suivantes : ($a1$ puis $a2$) et ($a2$ puis $a1$). En conséquence, l'algorithme doit vérifier que toutes les linéarisations sont solutions pour que le modèle de tâches le soit. Dans le cas où aucun plan n'est trouvé, le problème n'a pas de solution et aucune IHM ne peut être composée.

L'ALGORITHME PLACID

Nous proposons l'algorithme PLACID pour résoudre le problème de la Composition Dynamique de Services IHM. L'objectif de l'algorithme est de réaliser l'objectif de l'utilisateur (la tâche but).

Quelques notations sont d'abord introduites pour décrire l'algorithme PLACID.

Notations

Une tâche peut être une méthode ou un opérateur. La racine du modèle de tâches (plan) à calculer est appelée la tâche but t_u . En pratique, la projection ρ est un ensemble d'actions représentant le modèle de tâches. $\rho(t_u)$ est la projection de racine t_u où t_u est la tâche but.

Une fois $\rho(t_u)$ calculé, PLACID est capable de calculer les préconditions et les effets de t_u . Plus formellement, pour calculer les préconditions et les effets de la tâche t_u , deux cas sont à considérer si t_u est une action ou une méthode.

- Cas 1 (t_u est une action a) : si $\rho(t_u) = \{a\}$ est solution, alors $precond(t_u) = precond(a)$, $add(t_u) = add(a)$ et $del(t_u) = del(a)$.
- Cas 2 (t_u est une méthode m) : soit t_1, \dots, t_n les sous-tâches de t_u dans $\rho(t_u)$. Si $\rho(t_u)$ est une solution, alors $precond(t_u) = precond(t_1) \cup \dots \cup precond(t_n)$, $add(t_u) = add(t_1) \cup \dots \cup add(t_n)$ et $del(t_u) = del(t_1) \cup \dots \cup del(t_n)$. Intuitivement, les préconditions (resp. effets) de t_u sont l'union des préconditions (resp. effets) de ses sous-tâches.

Ainsi, l'algorithme va diviser le problème en sous-problèmes grâce aux méthodes disponibles. Une fois que l'algorithme a trouvé que toutes les sous-tâches sont des solutions, il doit vérifier que les sous-tâches puissent être appliquées soit en séquence soit en parallèle.

Pour vérifier si deux tâches peuvent être composées en parallèle, nous adaptons la notion de conflit déjà utilisée en planification PDDL. Si deux tâches ne sont pas en conflit, elles sont indépendantes. Plus formellement, deux tâches t_1 et t_2 sont indépendantes si et seulement si :

- $del(t_1) \cap (precond^+(t_2) \cup add(t_2)) = \emptyset$ et
- $add(t_1) \cap (precond^-(t_2) \cup dell(t_2)) = \emptyset$ et
- $del(t_2) \cap (precond^+(t_1) \cup add(t_1)) = \emptyset$ et
- $add(t_2) \cap (precond^-(t_1) \cup dell(t_1)) = \emptyset$.

Intuitivement, deux tâches sont indépendantes si les effets d'une tâche ne sont pas en conflit avec les préconditions de l'autre. Si les tâches en parallèle sont applicables et ne sont pas en conflit, alors t_u est réalisable.

Algorithme

L'algorithme adopte la stratégie de diviser pour régner (divide and conquer). En effet, en partant de l'objectif de l'utilisateur (t_u), l'algorithme va récursivement décomposer cette tâche en sous-tâches grâce aux méthodes disponibles pour obtenir un arbre où les feuilles sont des actions. Si toutes les tâches sont applicables et que les tâches parallèles ne sont pas en conflit, alors l'algorithme trouve une solution : le modèle de tâches permettant à l'utilisateur d'atteindre son objectif.

Soit $P = (s_0, t_u, O, M)$ un problème où s_0 est l'état initial, t_u est la tâche but à effectuer (une tâche est une méthode ou une action, par ex : « get medical assistance »), O et M sont les actions et les méthodes disponibles. Pour calculer l'ensemble des linéarisations (une solution), l'algorithme cherche une linéarisation et vérifie s'il n'y a pas de conflit entre les tâches en parallèle. Plus formellement, $\rho(t_u)$ est une solution à P si et seulement si :

- Cas 1 (t_u est une action a) : a est applicable à s_0 . Alors $\rho(t_u) = \{a\}$.
- Cas 2 (t_u est une méthode complètement instanciée m) : soit $\{t_1, \dots, t_n\} \in subAct(m)$; il y a deux cas si la méthode est de type séquence ou parallélisme :
 - Cas 2a (m est de type séquence) : m est applicable à s_0 et $\forall t_i \in subAct(m)$, $\rho(t_i)$ est solution à $P_i = (\gamma(s_0, t_{i-1}), t_i, O, M)$ avec $\gamma(s_0, t_{i-1}) = s_0$ si $i = 1$. Intuitivement, si t_u est une séquence, une solution est une linéarisation classique de ses sous-tâches.
 - Cas 2b (m est de type parallélisme) : m est applicable à s_0 et $\forall t_i \in subAct(m)$, $\rho(t_i)$ est solution à $P_i = (s_0, t_i, O, M)$ et $\forall t_i, t_j \in subAct(m)$, $i \neq j$, t_i et t_j sont indépendantes. Intuitivement, l'algorithme vérifie que toutes les sous-tâches en parallèle sont applicables à s_0 et qu'il n'y a pas de conflit entre elles. L'absence de conflit est vérifié pour permettre à l'utilisateur de sélectionner lui-même l'ordre dans lequel il veut effectuer les sous-tâches.

IMPLEMENTATION SUR UN EXEMPLE APPLICATIF

PLACID est illustré sur un exemple applicatif : Victor a besoin d'une assistance médicale durant ses vacances. Premièrement l'exemple applicatif est présenté. Puis, l'implémentation des opérateurs et des méthodes nécessaires est décrite. Finalement, les spécificités de PLACID sont discutées.

Exemple applicatif

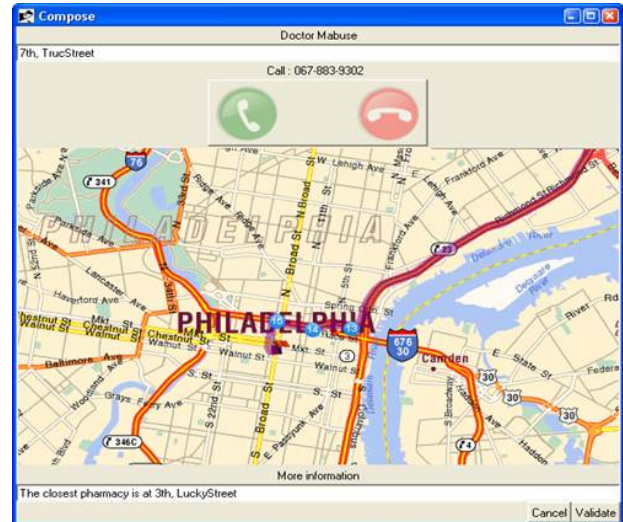


Figure 3. IHM composée pour que Victor ait une assistance médicale.

Victor vit à New York. Soudainement, il ne se sent pas bien alors qu'il est en vacances à Philadelphie. Il a besoin d'assistance médicale. Heureusement, il a un système pour l'aider. Il spécifie son objectif « Je veux une assistance médicale » sur le mur numérique de l'hôtel. En retour, PLACID compose dynamiquement un plan en termes de services IHM. Chaque service IHM décrit les services fournis par des composants disponibles. Le plan exprime donc l'IHM qui doit être composée. Par exemple, l'IHM composée permet d'abord à Victor de choisir son médecin. Une fois le médecin choisi, Victor peut appeler le médecin choisi « Dr Mabuse » (figure 3 ; le numéro est pré-composé – voir le haut de la fenêtre), être guidé jusqu'au cabinet médical (figure 3 ; voir la carte) et avoir l'adresse du pharmacien le plus proche (figure 3 ; voir l'adresse au bas de la fenêtre).

Implémentation

Pour implémenter cet exemple applicatif, il est nécessaire de disposer des composants IHM fournissant des services IHM décrits par des opérateurs et des méthodes. Les opérateurs et méthodes suivants sont implémentés pour notre exemple :

Opérateur *Choose.the.doctor* : les paramètres sont un utilisateur ($?u$), une plate-forme ($?p$). Le service IHM requiert que l'utilisateur ait une plate-forme d'exécution : $has ?u ?p$. Une fois que l'utilisateur a sélectionné le médecin, un prédicat est ajouté à l'état du monde : $isChosen ?u d$.

Opérateur *Call.the.office* : les paramètres sont un utilisateur ($?u$), une plate-forme ($?p$) et une personne à appeler ($?pp$). Le service IHM requiert que la plate-forme ait des capacités téléphoniques : $isPhone ?p$. Le fait que la personne pp ait appelé quelqu'un est ajouté à l'état du monde : $isCalled ?u ?pp$.

Opérateur *Find.route.info* : les paramètres sont un utilisateur ($?u$) et une plate-forme ($?p$). Le service IHM requiert que l'utilisateur ait une plate-forme connectée à Internet : $isInternet ?p$.

Opérateur *Find_nearest_chemist* : les paramètres sont un utilisateur (?u), une plate-forme (?p) et un pharmacien (?o). Le service IHM requiert que l'utilisateur ait une plate-forme (*has ?u ?p*) et que l'adresse du pharmacien puisse être affichée sur cette plate-forme : *isDisplayable ?o ?p*. L'état est enrichi de : *isDisplayed ?p ?o*.

Méthode *Get_medical_assistance* : les paramètres sont un utilisateur (?u) et une plate-forme (?p). Le service IHM requiert que l'utilisateur ait une plate-forme : *has ?u ?p*. Les sous-tâches sont « *Choose_the_doctor* » et « *Contact_the_doctor* » en séquence.

Méthode *Contact_the_doctor* : les paramètres sont un utilisateur (?u) et une plate-forme (?p). Les sous-tâches sont « *Call_the_office* », « *Find_route_info* » et « *Find_nearest_chemist* » en parallèle. Ce service IHM requiert que l'utilisateur ait une plate-forme d'exécution avec un écran large (*isLargeScreen ?p*).

A partir de ces opérateurs, ces méthodes et de l'objectif de l'utilisateur, l'algorithme calcule la solution (plan) présentée dans la figure 4. En effet, pour vérifier si la méthode « *Get_medical_assistance* » est solution, l'algorithme vérifie si la linéarisation de « *Choose_the_doctor* » et « *Contact_the_doctor* » est solution. « *Choose_the_doctor* » est applicable sur s_0 et produit s_1 . Pour vérifier si la méthode « *Contact_the_doctor* » est applicable sur s_1 et produit s_5 , l'algorithme vérifie si les actions « *Call_the_office* », « *Find_route_info* » et « *Find_nearest_chemist* » peuvent être appliquées sur s_1 et si les actions sont indépendantes. Si elles ne sont pas en conflit, s_5 peut être calculé à partir d'une linéarisation de ces actions.

Discussion

Une fois calculé, le plan peut être mis en correspondance avec les composants IHM. En effet, chaque service IHM est fourni par un composant IHM. Une méthode exprime une manière de composer les sous-composants IHM. Par exemple, la figure 5 affiche une IHM qui peut être composée à l'aide du plan de services IHM par le prototype COMPOSE [8]. Les composants correspondant aux tâches abstraites sont en charge de composer ses sous-composants. Par exemple, le composant permettant « *Contact_the_doctor* » est une frame qui va composer verticalement les IHM de « *Call_the_office* », « *Find_route_info* » et « *Find_nearest_chemist* ».

Par rapport au planificateur classique de services fonctionnels qui ne fournissent pas les méthodes en sortie, PLACID calcule un arbre contenant les méthodes et les actions. En effet, les méthodes expriment la composition, i.e., la manière de composer les composants IHM fournis par les services IHM. Par exemple, la méthode « *Contact_the_doctor* » exprime une fenêtre qui va contenir verticalement les composants IHM fournissant les services IHM (opérateurs) « *Call_the_office* », « *Find_route_info* » et « *Find_nearest_chemist* ».

Par rapport au planificateur classique de services fonctionnels, PLACID compose des services IHM en parallèle.

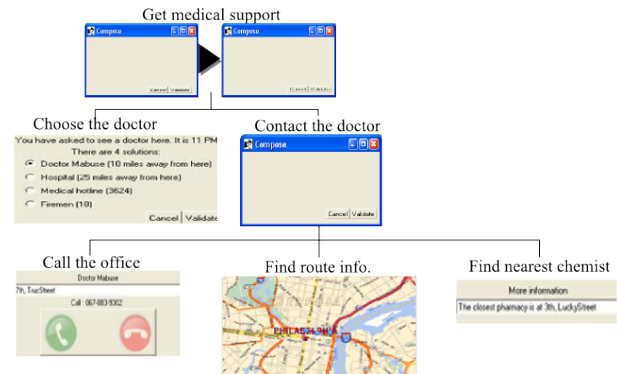


Figure 5. Composants IHM fournis par leur services IHM respectifs.

En effet, si deux services IHM ne sont pas en conflit, alors les deux composants IHM correspondant peuvent être composés dans la même temporalité (dans une même fenêtre ou onglet). Par exemple, un système de composition ne doit pas composer deux composants IHM qui nécessiteraient tous les deux les mêmes haut-parleurs.

Pour produire un système capable de composer dynamiquement une IHM, ce dernier doit disposer des composants IHM, les services IHM correspondant et d'un lien entre les deux. Un composant IHM peut être réutilisé pour fournir plusieurs services IHM. Par exemple, le composant IHM qui compose des IHM verticalement dans une même fenêtre sera souvent réutilisé.

Jusqu'à maintenant, PLACID a été appliqué à un exemple applicatif. Il est nécessaire d'étudier les performances de l'algorithme pour réaliser une composition d'IHM dynamique. Cette étude nécessite de réaliser des Benchmarks de services IHM qui n'existent pas actuellement.

Grâce à sa spécificité unique dans la littérature de retourner un arbre au lieu d'une séquence d'actions, PLACID peut également être utilisé à la conception comme générateur du modèle de tâches pour aider le concepteur à modéliser la tâche utilisateur. Cela est un intéressant effet de bord car les concepteurs d'IHM ne sont pas très familiarisés avec la modélisation en tâches utilisateur. En effet, la conception de TM est une tâche qui peut être accélérée et facilitée par l'utilisation de PLACID.

CONCLUSION ET PERSPECTIVES

Cet article a présenté une solution pour composer dynamiquement des IHM. La contribution principale est une formalisation de l'algorithme PLACID qui permet de calculer un plan à partir de services IHM. PLACID couvre la composition séquentielle et parallèle de services IHM (tâches) et qui produit un plan sous la forme d'un arbre de tâches. En effet, des services IHM peuvent être composés en séquence ou dans une même fenêtre. Pour vérifier si deux services peuvent être composés en parallèle, PLACID vérifie si les services IHM ne sont pas en conflit. Comme perspective, nous envisageons d'évaluer les performances de l'algorithme. Nous envisageons également d'évaluer PLACID comme outil d'aide à la conception rapide de modèle de tâches.

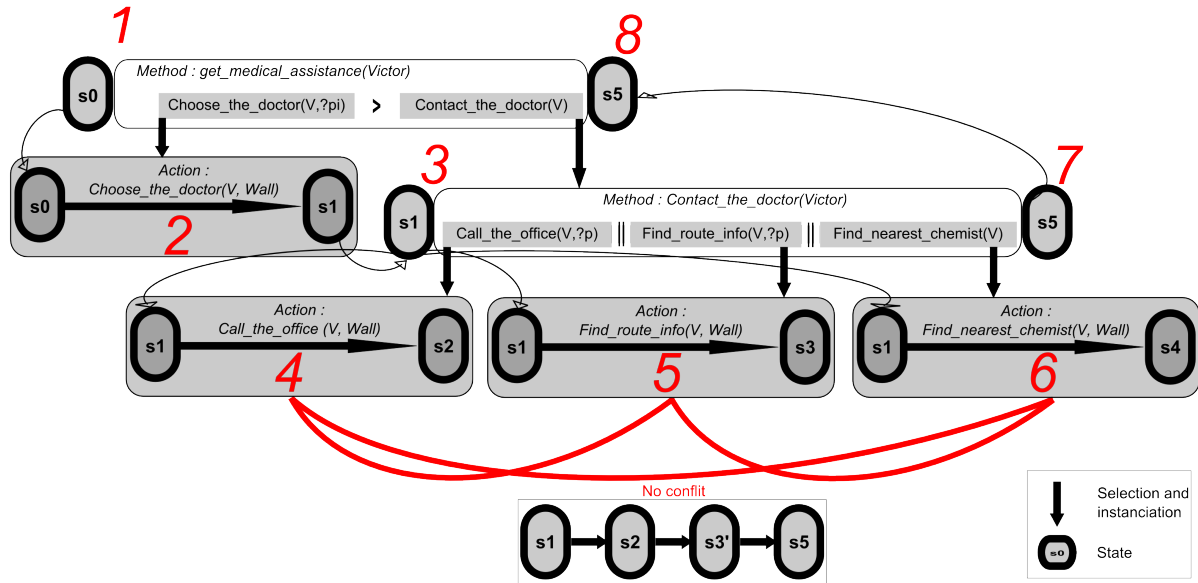


Figure 4. Exécution de PLACID sur l'exemple applicatif.

BIBLIOGRAPHIE

- Bidot J., Goumopoulos C. & Calemis I. Using ai planning and late binding for managing service workflows in intelligent environments. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, IEEE (2011), 156–163.
- Brel C., Dery-Pinna A.-M., Renevier-Gonin P. & Riveill M. Ontocompo: a tool to enhance application composition. In *Human-Computer Interaction-INTERACT 2011*, Springer (2011), 588–591.
- Bucchiarone A. & Gnesi S. A survey on services composition languages and models. In *Proceedings of International Workshop on Web Services Modeling and Testing (WS-MaTe2006)* (2006), 51–63.
- Calvary G., Coutaz J., Thevenin D., Limbourg Q., Bouillon L. & Vanderdonck J. A unifying reference framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (2003), 289–308.
- Coutaz J. Software architecture modeling for user interfaces. *Encyclopedia of software engineering* (1993).
- Erol K., Hendler J. & Nau D. S. Htn planning: Complexity and expressivity. In *AAAI*, vol. 94 (1994), 1123–1128.
- Feldmann M., Martens F., Berndt G., Spillner J. & Schill A. Rapid development of service-based interactive application using service ANNOTATIONS. 319–322.
- Gabillon Y., Petit M., Calvary G. & Fiorino H. Automated planning for user interface composition. In *Proceedings of the 2nd International Workshop on Semantic Models for Adaptive Interactive Systems: SEMAIS'11 at IUI 2011 conference*, Springer HCI (2011).
- Ghallab M., Nau D. & Traverso P. *Automated planning: theory & practice*. Morgan Kaufmann, 2004.
- Heinroth T., Kameas A., Pruvost G., Seremeti L., Bellik Y. & Minker W. Human-computer interaction in next generation ambient intelligent environments. *Intelligent Decision Technologies* 5, 1 (2011), 31–46.
- Joffroy C., Caramel B., Dery-Pinna A.-M. & Riveill M. When the functional composition drives the user interfaces composition: process and formalization. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, ACM (2011), 207–216.
- Klusch M., Gerber A. & Schmidt M. Semantic web service composition planning with owls-xplan. In *Proceedings of the AAAI Fall Symposium on Semantic Web and Agents, Arlington VA, USA*, AAAI Press (2005).
- Kritikos K., Plexousakis D. & Paternò F. Task model-driven realization of interactive application functionality through services. *ACM Transactions on Interactive Intelligent Systems (TiIS)* 3, 4 (2014), 25.
- McIlraith S. & Son T. Adapting golog for composition of semantic web services. In *Principles of Knowledge representation and reasoning-International Conference*, Citeseer (2002), 482–496.
- Medjahed B. *Semantic web enabled composition of web services*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, 2004.
- Motti V. G. & Vanderdonck J. A computational framework for context-aware adaptation of user interfaces. In *Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on*, IEEE (2013), 1–12.
- Nichols J., Rothrock B., Chau D. H. & Myers B. A. Huddle: automatically generating interfaces for systems of multiple connected appliances. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM (2006), 279–288.
- Paternò F., Mancini C. & Meniconi S. Concurtasktrees: A diagrammatic notation for specifying task models. In *Human-Computer Interaction INTERACT'97*, Springer (1997), 362–369.
- Paternò F., Santoro C. & Spano L. D. Engineering the authoring of usable service front ends. *Journal of Systems and Software* 84, 10 (2011), 1806–1822.
- Pietschmann S., Voigt M., Rümpel A. & Meißner K. Cruise: Composition of rich user interface services. In *Web Engineering*. Springer, 2009, 473–476.
- Rao J., Küngas P. & Matskin M. Application of linear logic to web service composition. In *IWCS*, L.-J. Zhang, Ed., CSREA Press (2003), 3.
- Rao J. & Su X. A survey of automated web service composition methods. *Semantic Web Services and Web Process Composition* (2005), 43–54.
- Sirin E., Parsia B., Wu D., Hendler J. & Nau D. Htn planning for web service composition using shop2. *Web Semantics: Science, Services and Agents on the World Wide Web* 1, 4 (2004), 377–396.
- Thevenin D. & Coutaz J. Plasticity of user interfaces: Framework and research agenda. In *Proceedings of INTERACT*, vol. 99 (1999), 110–117.

Un système d'aide et de suivi des tâches utilisateur dans un environnement ambiant

Asma Gharsellaoui
LIMSI-CNRS, Université
Paris Sud
91403 Orsay Cedex, France
asma.gharsellaoui@limsi.fr

Yacine Bellik
LIMSI-CNRS, Université
Paris Sud
91403 Orsay Cedex, France
yacine.bellik@limsi.fr

Christophe Jacquet
Dép.informatique-Supélec
91192 Gif-sur-Yvette Cedex,
France
christophe.jacquet@supelec.fr

RÉSUMÉ

Les modèles de tâches existants sont généralement statiques (non utilisés au moment de l'exécution) et exploités uniquement pour la conception ou l'évaluation prédictive des systèmes interactifs. Nous proposons d'utiliser le modèle de tâches au moment de l'exécution, afin de suivre les actions de l'utilisateur, vérifier qu'il n'a pas fait d'erreurs et lui procurer de l'aide en cas de besoin. Nous présentons un modèle de tâches adapté aux environnements ambiants qui attribue dynamiquement des états aux tâches au moment de l'exécution. Nous décrivons également le fonctionnement d'un système de suivi et d'assistance qui exploite notre modèle de tâches dynamique. Nous présentons ensuite une validation de notre système à travers une simulation qui montre comment les interactions avec le modèle de tâches à l'exécution permettent de produire un système dynamique, qui prend en considération le contexte et fournit une aide à l'utilisateur pour la réalisation de ses tâches quotidiennes.

Mots Clés

Modélisation des tâches utilisateurs ; intelligence ambiante ; interaction dans les environnements ambiants.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces.

INTRODUCTION

Le terme intelligence ambiante (AmI) a été introduit initialement par la Commission Européenne en 2001 [6], [25], bien que le concept en lui-même remonte au début des années 90 [34] [35]. Il désigne une vision de l'environnement quotidien de l'utilisateur dans lequel la technologie est omniprésente et discrète [24]. Un système d'intelligence ambiante est un système interactif dans lequel des objets intelligents embarqués dotés de capacités de calcul et de communication coopèrent afin de répondre au mieux aux besoins des utilisateurs humains [26] [11]. Les systèmes ambiants ont pour objectif de percevoir et d'agir sur l'environnement physique, de façon à s'intégrer dans les activités quotidiennes des utilisateurs. À partir

des données issues de capteurs sur l'état de l'environnement (les objets manipulés, la position de l'utilisateur, la tâche en cours de réalisation...), nous soutenons que le système doit être capable de suivre le déroulement des tâches utilisateur et de l'assister en cas de besoin. Dans la première section de cet article, nous nous intéressons aux spécificités de la modélisation des tâches dans les environnements ambiants. La deuxième section présente le modèle de tâches que nous proposons, et qui est exploitable à l'exécution. Le système d'aide et de suivi est ensuite présenté en détaillant ses stratégies d'intervention. Dans une dernière partie, nous décrivons le simulateur qui a été développé et nous détaillons son fonctionnement à travers le déroulement d'un scénario dans les environnements ambiants.

LA MODÉLISATION DES TÂCHES DANS LES ENVIRONNEMENTS AMBIANTS

Besoins de la modélisation des tâches dans les environnements ambiants

Un modèle de tâches décrit les actions destinées à être effectuées afin d'atteindre les objectifs de l'utilisateur [27] et les différentes manières de les atteindre [18]. La modélisation des tâches vise en particulier à construire un modèle qui décrit précisément les relations entre les différentes tâches [21]. Ces modèles sont employés pour des finalités variées telles que l'aide à la compréhension de l'activité, la conception de programmes de formation [2] [3] [1] [22], la conception et l'évaluation des interfaces homme-machine [4] [13]. À cet effet, plusieurs catégories d'utilisateur sont amenés à produire ces modélisations, des utilisateurs novices (par exemple les étudiants lors des formations) aux experts. Plusieurs recherches [17] [19] [23] se sont intéressées à la construction des modèles de tâches et ont mesuré les erreurs syntaxiques et sémantiques et le temps de réalisation du modèle. Ces études ont montré que la production d'un modèle de tâches semble être plus complexe que la compréhension d'un modèle déjà construit. Les concepteurs d'outils de modélisation doivent tenir compte de cette contrainte en réduisant la charge de travail mentale associée à la modélisation et ne pas surcharger visuellement les modèles. Dans les environnements ambiants, la modélisation des tâches présente certains besoins spécifiques [7]. Le modèle doit premièrement inclure la possibilité d'étiqueter une tâche selon des contraintes spatiales, deuxièmement permettre de spécifier les ressources physiques nécessaires à sa réalisation, troisièmement déterminer le niveau de granularité de la tâche à modéliser.

La principale caractéristique d'un tel modèle est qu'il doit être dynamique. En effet, ce modèle doit pouvoir être mis à jour au moment de l'exécution afin de suivre le déroulement des tâches au moment où elles sont accomplies.

Exploitation du modèle de tâches à l'exécution

Un grand nombre de modèles de tâches a été développé, en particulier dans le contexte des interfaces graphiques : HTA [2], GTA [33], CTT [20], UAN [30], TKS [10] [9], DIANE + [32], TOOD [15] et K-MAD [5]. Une étude comparative décrite dans [13] a montré que certaines propriétés doivent être prises en charge dans les modèles comme : la description des buts, une structure hiérarchique pour la modélisation des tâches, l'intégration d'opérateurs temporels décrivant le séquençement des tâches, et les détails des objets et des actions qui permettront la modélisation des interfaces utilisateurs. Cette étude a montré que plus les modèles de tâches sont expressifs plus ils deviennent complexes. Du moins complexe et le moins expressif au plus riche on trouve : HTA, MAD, GTA, TKS, CTT, Diane+, TOOD.

Dans le domaine de la génération automatique d'interfaces utilisateur, Klug et Kangasharju [12] présentent un modèle de tâches exécutable exploité pour générer une interface graphique dynamique composée de blocs d'interface préprogrammés. Ils étendent quelques notations de CTT pour permettre l'exécution dynamique du modèle de tâches en runtime en donnant quelques états dynamiques aux tâches feuilles.

Dans [36], les auteurs comparent les concepts de tâches dans deux langages de modélisation (développés comme des extensions d'un même modèle de tâches, CTT) qui sont UsiXML [14] et CTML [37]. UsiXML permet la création de différentes versions d'une interface adaptées à différents contextes. Il réutilise les opérateurs temporels standard de CTT et propose d'affiner l'opérateur « Choice » [29] [28]. On peut cependant noter que si CTT supporte l'utilisation informelle de pré-conditions des tâches, cela n'est pas exprimable dans UsiXML. CTML, quant à lui, est un langage de modélisation de tâches coopératives qui a été développé pour les environnements ambiants. Il permet de prendre en considération la dépendance de la localisation des objets et la modélisation des périphériques invoqués. L'avantage de CTML est qu'il utilise des pré-conditions et des effets évalués sémantiquement.

Nous avons constaté que CTML mis à part, aucun des modèles de tâches existants ne correspondait parfaitement aux exigences de la modélisation des tâches dans les environnements ambiants. Mais ce dernier n'exploite pas les performances temporelles des tâches et ne permet pas d'assister l'utilisateur. De plus l'éditeur et le simulateur de CTML ne sont pas disponibles publiquement, ce qui empêche sa réutilisation.

Plutôt que de proposer un nouveau modèle de tâches ex nihilo, nous avons préféré partir d'un modèle existant et l'étendre. Notre choix s'est porté sur le modèle CTT. Il intègre un ensemble riche d'opérateurs temporels ; il offre

la possibilité de définir des pré-conditions et des post-conditions d'une tâche selon une certaine syntaxe (que nous nous proposons de raffiner par la suite) et il propose un éditeur accessible qui permet de saisir le modèle de tâches et de vérifier sa cohérence (CTT Environment ou CTTE) [16].

UN MODÈLE DE TÂCHES EXPLOITABLE EN RUNTIME

Dans cette section nous présentons les différentes extensions apportées à CTT dans notre modèle.

Transformation de l'arbre des tâches

Partant d'un modèle de tâches CTT, nous commençons par le transformer de façon à obtenir un modèle de tâches qui se présente uniquement sous la forme d'un arbre binaire sans liens horizontaux entre les tâches : les liens horizontaux de CTT, qui représentent les opérateurs temporels, donnent lieu à de nouvelles tâches abstraites intermédiaires correspondant à ces opérateurs. Cette transformation est introduite dans le but de simplifier l'algorithme de suivi du déroulement qui n'aura plus qu'à s'appliquer sur un arbre binaire classique. Nous distinguons deux types de tâches dans notre arbre binaire : les tâches concrètes et les tâches abstraites. Les tâches concrètes sont les feuilles de l'arbre. Elles représentent des tâches élémentaires qui peuvent être détectées par les capteurs de l'environnement, lors de leurs exécutions. Les tâches abstraites reflètent le comportement d'un opérateur temporel reliant deux sous-tâches, comme illustré sur la figure 1 :

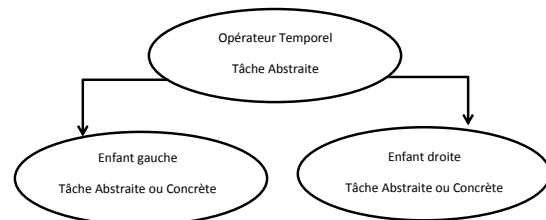


Figure 1. Exemple de décomposition de tâches.

La figure 2 montre un exemple simple de la façon dont un modèle CTT (conçu avec CTTE) est transformé en un arbre binaire.

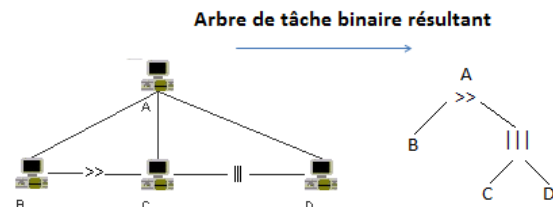


Figure 2. Arbre de tâches binaire résultant de la transformation.

Annotation des tâches élémentaires

Une tâche élémentaire a un certain nombre de propriétés qui peuvent être soit statiques (une fois établie lors de la phase de conception, la valeur de la propriété ne pourra plus être modifiée lors de la phase d'exécution), soit dynamiques (la valeur de la propriété peut évoluer au moment

de l'exécution en fonction des informations provenant des capteurs).

Les propriétés statiques d'une tâche sont de deux types.

- On retrouve certaines propriétés qui ont été déjà introduites dans CTT et qui sont :
 - Son identifiant ou son nom.
 - Son importance (élevée, moyenne, basse).
 - Son caractère obligatoire : détermine si la réalisation de la tâche est indispensable pour atteindre un objectif donné.
 - Sa durée maximum de réalisation.
 - Sa durée minimum de réalisation.
- On aura aussi un certain nombre de propriétés que nous avons raffinées ou introduites pour les besoins propres de notre modèle et qui sont :
 - Service : représente un service invoqué par la tâche. On introduit ici la notion de service abstrait pour désigner un processus qui représente une suite finie de services destinée à atteindre un but donné.
 - Pré-condition : Une condition qui doit exister ou doit être établie avant que la tâche ne puisse se produire.
 - Post-condition : Constitue un ensemble de conditions dont on sait qu'elles doivent être vérifiées après l'exécution de la tâche. Les pré/post conditions peuvent contenir une information sur principalement trois grandes catégories de composantes à savoir :
 - l'environnement, par exemple la température
 - l'état des dispositifs du système
 - l'utilisateur : ses profils statique et dynamique. Par exemple le sexe sera une caractéristique statique alors que la position de l'utilisateur sera une caractéristique dynamique puisqu'elle peut changer au cours de l'utilisation du système.

Les pré-conditions utilisées dans notre modèle sont écrites selon une syntaxe que nous avons définie et sont évaluées au moment de l'exécution. Une pré-condition est une formule en logique propositionnelle (dont les différentes parties sont reliées par les opérateurs logiques « OU », « ET » et « NON »).

Les propriétés dynamiques d'une tâche que nous avons introduites sont :

- État de la tâche (inactive, possible, réalisable, active, suspendue, réalisée, arrêtée) : déterminé en utilisant les relations temporelles reliant les différentes tâches (voir ci-dessous).
- État des pré-conditions : il est modifié lorsqu'un événement se produit dans l'environnement.
- État des post-conditions : il est modifié suite à l'exécution d'une tâche.

Les opérateurs temporels

Les opérateurs temporels utilisés dans notre modèle sont ceux de CTT. Nous listons dans ce qui suit les différents opérateurs, selon leur ordre de priorité, en commençant par le plus prioritaire : Choice [], Order Independent |=, Interleaving ||, Synchronization |[], Disabling [>, Suspend-Resume | >, Sequential Enabling >>.

Les états des tâches

Comme nous proposons un modèle de tâches dynamique, l'état des tâches est recalculé en continu en fonction des

changements survenus dans l'environnement. Les états possibles d'une tâche sont :

- INACTIVE : état initial ; la tâche n'a pas encore été effectuée.
- POSSIBLE : cela signifie que temporellement, la tâche peut être effectuée dès que ses pré-conditions seront satisfaites.
- RÉALISABLE : la tâche peut être effectuée (ses pré-conditions sont satisfaites).
- ACTIVE : la tâche est en cours d'exécution.
- SUSPENDUE : cette tâche était en cours d'exécution (Active) lorsqu'une autre tâche a démarré et l'a suspendue. Sa réalisation sera reprise une fois que la nouvelle tâche sera finie. Le même cas s'applique pour une tâche pouvant être réalisée (Possible ou Réalisable) qui ne pourra être démarrée que lorsque la tâche qui l'a suspendue sera terminée.
- ARRÊTÉE : l'exécution de la tâche a été interrompue par une autre et ne pourra pas être reprise. Notons toutefois que contrairement à la suspension l'arrêt d'une tâche ne peut s'appliquer qu'aux tâches qui ont été effectivement démarrées (Actives) et non aux tâches en attente de démarrage.
- RÉALISÉE : la tâche a été réalisée entièrement.

Les états des tâches et les transitions entre eux sont représentés dans la figure 3.

UN SYSTÈME D'AIDE ET DE SUIVI BASÉ SUR LE MODÈLE DE TÂCHES

Un scénario dans les environnements ambiants

Afin de faciliter la compréhension de l'approche, nous présentons un scénario que nous utiliserons dans la suite de l'article :

Il est 10h00. Jean s'apprête à recevoir des amis dans 2 heures. Il peut soit commencer par la préparation du déjeuner soit par le nettoyage de la maison. Chacune de ces deux tâches peut être interrompue par un appel téléphonique ou par l'arrivée de quelqu'un qui frappe à la porte.

Pour préparer le déjeuner, il doit commencer par la cuisson des pâtes : il doit tout d'abord prendre la casserole dans le placard, la remplir d'eau et la mettre sur le feu. Dix minutes plus tard, une fois que l'eau est bouillante, il devra mettre les pâtes dans la casserole, les laisser cuire pendant 15 minutes puis les égoutter.

Pendant que les pâtes sont en train de cuire, il devra préparer la sauce en commençant par éplucher un oignon. À ce moment, nous supposons que le téléphone fixe qui est dans le salon commence à sonner. Jean devra alors suspendre sa tâche en cours pour répondre à l'appel. Une fois la communication terminée, il devra revenir à la cuisine, couper l'oignon en petits morceaux, prendre la poêle du placard et y mettre l'oignon. Il devra ensuite ajouter l'huile et mettre le mélange dans la cocotte puis cinq minutes plus tard y verser le contenu d'une sauce préalablement achetée en grande surface et laisser cuire pendant dix minutes. Nous supposons que dix minutes plus tard le système détecte que la plaque de cuisson n'a pas été éteinte. Le système devra alors indiquer à Jean d'enlever la sauce de la plaque de cuisson et de l'éteindre. Jean devra ensuite mélanger les pâtes avec la sauce. Une

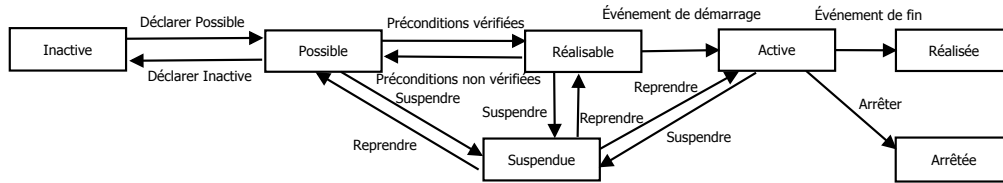


Figure 3. Les états possibles d'une tâche sous forme d'une machine à état finis.

fois le déjeuner prêt, il devra s'occuper du nettoyage de la maison.

Architecture globale de la solution

Le principe de notre approche est de comparer, au moment de l'exécution des tâches, les informations contenues dans le modèle de tâches à l'activité réelle de l'utilisateur afin de pouvoir lui apporter une assistance en cas de besoin. La figure 4 illustre cette approche.

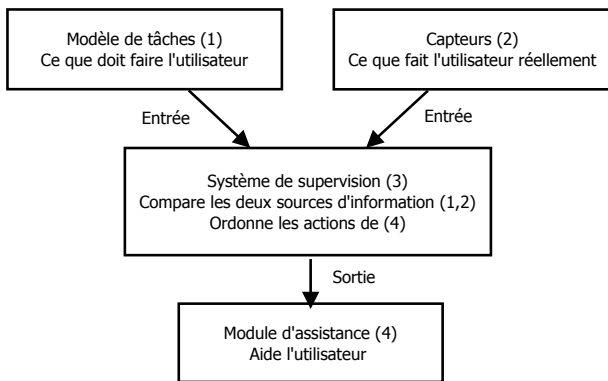


Figure 4. Approche globale.

Le système de supervision (3) reçoit en permanence des informations issues des capteurs (2) sur ce qui se passe réellement dans l'environnement. Ces informations servent à mettre à jour d'une part l'état des tâches (détection du démarrage et de la fin d'une tâche donnée) et d'autre part l'état des pré-conditions. Les valeurs reçues des capteurs influencent directement les valeurs des pré-conditions qui les utilisent comme des variables. Dans notre scénario, à titre d'exemple, la tâche « Sortir poêle du placard » a comme pré-condition « le placard est ouvert », cette pré-condition est vérifiée par le capteur posé au niveau de la porte du placard. En ce qui concerne le démarrage des tâches, nous ne traitons pas cette problématique pour le moment. Nous supposons que le démarrage est détecté en s'appuyant sur les travaux existants qui portent sur la reconnaissance de l'activité en cours, à titre d'exemple [8][31]. La détection de la fin des tâches est quant à elle effectuée grâce à l'utilisation des post-conditions affectées aux tâches. La vérification de ces post-conditions indique la fin de la tâche concernée. Par exemple la vérification de la fin de la tâche « éteindre poêle » revient à recevoir une information du capteur de température de la plaque de cuisson. Ces informations sont alors comparées à celles contenues dans le modèle de tâches (1) qui décrit ce que l'utilisateur devrait faire pour atteindre un but donné. Si le système constate que

l'utilisateur est en train de faire une action différente de ce qui a été prévu dans le modèle, il peut décider d'appeler le module d'assistance (4) afin d'informer l'utilisateur que telle sous-tâche n'a pas été faite correctement ou qu'elle a été omise. Dans le cas où l'utilisateur oublie d'éteindre la plaque, par exemple, le système détecte ce dysfonctionnement et appelle le module d'assistance qui indiquera à l'utilisateur de faire cette tâche.

Stratégies du système de supervision

Le système de supervision repose sur l'utilisation d'un module logiciel, le contrôleur de tâches, dont le but est de contrôler le déroulement correct des tâches. Ce contrôleur de tâches est lui-même composé de deux modules de contrôle différents ayant chacun un rôle particulier (figure 5) :

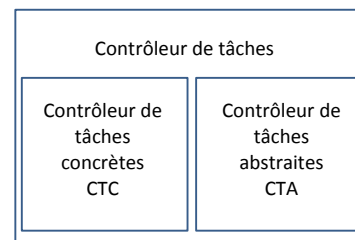


Figure 5. Schéma du contrôleur de tâche.

- Le contrôleur de tâches concrètes ou CTC : le CTC s'occupe du contrôle des tâches concrètes selon un algorithme commun à toutes les tâches concrètes.
- Le contrôleur de tâches abstraites ou CTA : le CTA gère le contrôle de l'ordonnancement des tâches selon des algorithmes différents dépendant du type de tâche abstraite, car chaque type de tâche abstraite correspond à un opérateur temporel différent.

Le CTC

Le CTC « surveille » les actions de l'utilisateur et invoque le module d'assistance dans un des deux cas suivants :

1. l'utilisateur fait une tâche qu'il ne devrait pas faire ;
2. l'utilisateur fait la bonne tâche mais pas comme il faut.

Notons qu'il existe un troisième cas où le module d'assistance peut être invoqué (« l'utilisateur ne fait pas une tâche qu'il devrait faire ») mais ce cas est pris en charge par le CTA. Dans ce qui suit nous allons expliciter comment le CTC gère chacun de ces deux cas. Considérons tout d'abord la figure 6 qui détaille le cycle de vie nominal d'une tâche (sans suspension ou arrêt). On définit « le temps d'attente maximum d'activation » comme la durée maximale prévue entre le moment où la tâche devient possible et le moment où

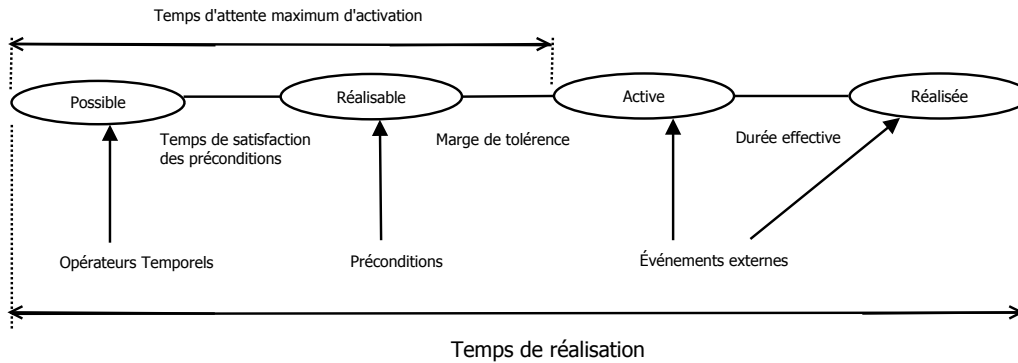


Figure 6. Cycle de vie nominal d'une tâche.

l'utilisateur commence concrètement sa réalisation. « Le temps de réalisation » correspond à la durée maximale prévisionnelle dédiée à la réalisation de la tâche. La figure 6 illustre ces différents états et les temps les séparant, décrivant ainsi l'ordonnancement normal des états d'une tâche qui va servir de référence pour le suivi de cette tâche.

1. L'utilisateur fait une tâche qu'il ne devrait pas faire.

Il s'agit du cas où l'utilisateur entame la réalisation d'une tâche alors que ce n'est pas le moment de la faire et/ou que ses pré-conditions ne sont pas vérifiées (tâche non réalisable). Le CTC peut détecter ce cas directement à partir des états dynamiques des tâches. Par exemple, dans notre scénario ce cas peut arriver dans le cas où Jean essaye de mettre les pâtes dans l'eau alors que celle-ci n'est pas encore bouillante.

2. L'utilisateur fait la bonne tâche mais pas comme il faut. Le CTC peut conclure qu'une tâche n'est pas (ou n'a pas été) effectuée correctement en se basant sur les deux critères suivants :

- L'effet : une tâche réalisée mais qui ne produit pas les effets attendus sera donc considérée comme une tâche n'ayant pas été effectuée correctement. Ce cas peut se présenter dans notre exemple dans le cas où l'utilisateur a effectué la tâche « éteindre la plaque de cuisson » alors que celle-ci reste à température élevée après un long moment.
- Les modalités de réalisation : si l'utilisateur est en train de réaliser une tâche, il doit forcément manipuler une ressource appartenant à la catégorie de ressources que nécessite la tâche. Dans le cas contraire, on peut dire que l'utilisateur est en train de réaliser la tâche mais pas de la manière attendue. Par exemple, le système peut détecter ce cas si Jean utilise le micro-onde pour la cuisson et pas la gazinière.

Le CTA

Le CTA a en charge la supervision temporelle globale des tâches selon deux axes :

- l'activation (correspond au cas où l'utilisateur ne fait pas une tâche qu'il devrait faire). Dans le scénario, cela se produit si l'utilisateur dépasse 10 minutes après l'ébullition de l'eau et ne commence pas la tâche de « cuisson des pâtes ».

- la réalisation. Ce cas peut arriver dans le scénario si la tâche de « cuisson des pâtes » dépasse les 15 minutes comme prescrit.

En ce qui concerne le contrôle de l'activation, le CTA informera l'utilisateur qu'il a du retard sur le démarrage de la tâche si le temps d'attente maximum d'activation qui lui a été associé est dépassé et si la tâche est toujours à l'état « Possible » ou « Réalisable » (voir figure 6). Pour le contrôle de la réalisation, nous expliquerons dans un premier temps, en détail, son fonctionnement dans le cas de tâches séquentielles et nous montrerons ensuite comment ce fonctionnement peut être étendu dans le cas des autres types d'ordonnancement possibles entre tâches (différents types d'opérateurs temporels).

On définit le temps de réalisation d'une tâche t comme étant la durée séparant l'instant où cette tâche est déclarée comme étant « Possible » jusqu'à l'instant où celle-ci atteint l'état « Réalisée ». Ici le CTA va s'intéresser à la durée de temps mise pour la réalisation de toutes les tâches de l'arbre. Pour cela on considère les deux fonctions $D_{MAX}(t)$ et $D_{eff}(t)$, qui définissent respectivement la durée maximale de la tâche t prévue par le concepteur, et la durée effective mise pour accomplir la tâche. Deux cas peuvent alors se présenter :

1. soit la tâche est réalisée dans un laps de temps inférieur ou égal à la durée maximale qui lui a été attribuée ($D_{MAX}(t) \geq D_{eff}(t)$) : dans ce cas on disposera d'une marge égale à $D_{MAX}(t) - D_{eff}(t)$.
2. soit la tâche est réalisée dans un laps de temps supérieur à la durée maximale qui lui a été attribuée ($D_{MAX}(t) < D_{eff}(t)$) : dans ce cas on signalera un retard sur la tâche t de durée $D_{eff}(t) - D_{MAX}(t)$

Si le cas 2 se présente, l'utilisateur a du retard par rapport à la durée prévue pour la réalisation de cette tâche. Le problème consiste à déterminer le moment où le CTA devra appeler le module d'assistance pour alerter l'utilisateur sur l'existence d'un risque de ne pas pouvoir terminer la réalisation des tâches dans le délai de temps restant. Considérons dans un premier temps, un arbre de tâches composé uniquement de n tâches séquentielles T_1 à T_n . En plus des contraintes temporelles que le concepteur peut spécifier au niveau de chaque tâche concrète, il est possible de poser une contrainte temporelle globale sur l'ensemble des tâches à réaliser (comme représenté sur la figure 7). La valeur $D_{MAX}(t)$ d'une tâche mère t qui se

décompose en un ensemble de tâches séquentielles peut être obtenue de deux manières différentes :

- spécifiée par le concepteur : le concepteur décide d'attribuer une durée de temps à un ensemble de tâches. Cette durée totale doit vérifier la condition suivante : $(\sum_{t' \in C} D_{MAX}(t')) \leq D_{MAX}(t)$, avec C l'ensemble des tâches concrètes situées sous la tâche mère t .
- le cas échéant calculée par cumul des durées prévisionnelles de ses tâches concrètes comme suit : $D_{MAX}(t) = \sum_{t' \in C} D_{MAX}(t')$;

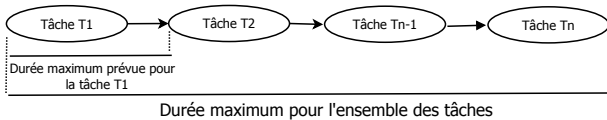


Figure 7. Temps maximum attribué à un ensemble de tâches.

Les données de notre problème sont les suivantes :

- n tâches à réaliser de façon séquentielle
- $D_{MAX}(i)$ pour i allant de 1 à n : durée maximum prévisionnelle pour la réalisation de la tâche i
- $D_{MIN}(i)$ pour i allant de 1 à n : durée minimum prévisionnelle pour la réalisation de la tâche i
- t_0 : temps où la tâche 1 (première tâche de la chaîne) passe à l'état possible
- t_c : temps courant
- k : nombre de tâches réalisées à l'instant t_c
- $D_{eff}(i)$ pour i allant de 1 à k : durée effective mise pour la réalisation de la tâche i

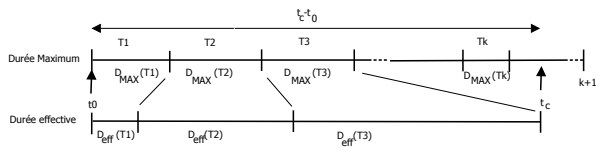


Figure 8. Temps maximum et effectifs d'un ensemble de tâches

Le CTA construit pour l'ensemble des tâches de la chaîne un tableau temporaire qui contiendra les temps de début, ainsi que les durées maximales et minimales prévisionnelles d'exécution des tâches. Pour calculer l'instant de début au plus tard d'une tâche, il se réfère aux durées maximales d'exécution $D_{MAX}(i)$ et pour calculer les instants de fin au plus tard il se réfère aux durées minimales d'exécution $D_{MIN}(i)$. Les instants de début sont calculés en partant de la première tâche de la chaîne alors que les instants de fin sont eux calculés à partir de la dernière tâche de la chaîne. Le résultat est illustré sur la table 1. Le CTA peut alors utiliser l'algorithme suivant :

1. calculer la durée $t_c - t_0$
2. repérer dans le tableau la tâche attendue T_{att} en se basant sur les temps de début au plus tard. Notons i_{att} son indice. La tâche attendue correspondra à celle ayant l'instant de début le plus grand entre les tâches ayant un instant de début au plus tard inférieure à $t_c - t_0$
3. deux cas de figures sont possibles :
 - (a) cas 1 : $k \geq i_{att}$ signifie que l'utilisateur est en avance par rapport au planning prévu : l'intervention du système n'est pas nécessaire
 - (b) cas 2 : $k < i_{att}$ signifie que l'utilisateur est en retard par rapport au planning prévu :

- i. cas 2a : $(\sum_{j=k+1}^n D_{MIN}(j)) \leq (D_{MAX} - (t_c - t_0))$, en d'autres termes la durée de temps restante peut suffire pour finir les tâches de $(k + 1)$ à n si on se base sur les durées maximales des tâches.
- ii. cas 2b : $(\sum_{j=k+1}^n D_{MIN}(j)) > (D_{MAX} - (t_c - t_0))$, en d'autres termes la durée de temps restantes ne suffira pas pour finir les tâches de $(k + 1)$ à n si on se base sur les durées minimales des tâches.

Deux stratégies d'intervention peuvent alors être considérées :

1. Stratégie intrusive ou stratégie pessimiste : dès qu'il y a un dépassement des durées maximales prédéfinies le système doit produire des alertes (le cas 2).
2. Stratégie moins intrusive ou stratégie optimiste : ici on considère que bien que l'utilisateur ait pris du retard sur la réalisation des tâches de 1 à k la situation peut être rattrapée, donc on vérifie d'abord si le temps restant peut suffire pour la réalisation des tâches restantes dans l'hypothèse où l'utilisateur ait des performances maximales. On ne génère donc des alertes système à l'utilisateur que dans le cas 2b.

Nous venons de voir la stratégie d'intervention du contrôleur de tâches abstraites appliquée à un ensemble de tâches séquentielles. Nous allons voir maintenant comment ceci s'étend aux autres types d'opérateurs temporels. Pour les calculs des temps de début et de fin au plus tard sur lesquels va se baser notre système de supervision, nous allons propager les calculs sur notre structure en arbre. Nous commençons par voir de près les opérateurs temporels restants. Nous distinguons deux classes d'opérateurs :

1. Opérateur temporel à une seule possibilité de réalisation : l'opérateur « Synchronisation ». Il existe une seule manière possible pour réaliser les deux tâches : démarrer leur réalisation en même temps et la finir en même temps. La durée maximale (ou minimale) pour une tâche mère de ce type est approximée à la durée maximale (ou minimale) prévue pour ces deux tâches filles.
2. Opérateurs temporels avec un choix qui se fait en runtime, qui eux même peuvent être partagés en deux groupes :
 - (a) ceux qui ont leur deux tâches filles obligatoires : « Interleaving » et « Order Independent ». La durée maximale (ou minimale) pour une tâche mère de ce type est approximée à la somme des durées maximales (ou minimales) prévues pour ces deux tâches filles.
 - (b) ceux qui n'ont qu'une tâche obligatoire et l'autre tâche optionnelle (pas forcément réalisée) qui à leur tour peuvent être partagé en deux groupes :
 - i. nous ne connaissons pas par avance quelle tâche sera réalisée : cas de l'opérateur « Choice ». La durée maximale d'une tâche mère de ce type est approximée à la durée maximale prévue pour sa tâche fille qui prend le plus de temps (de même pour l'approximation de la durée minimale mais cette fois

Tâche	Début au plus tard	Fin au plus tard
T_1	0	...
...
T_k	$D_{MAX}(1) + \dots + D_{MAX}(k)$	$D_{MAX} - \dots - D_{MAX}(k + 1)$
...
T_n	$D_{MAX}(1) + \dots + D_{MAX}(n - 1)$	D_{MAX}

Table 1. Calcul des instants de début et de fin au plus tard

en considérant les durées minimales prévues pour les deux tâches filles).

- ii. nous connaissons par avance quelle tâche sera réalisée (partiellement ou entièrement mais nous sommes sûrs de commencer avec cette tâche) : « Suspend Resume » et « Disabling ». Ici pour approximer la durée maximale ou minimale de la tâche mère deux stratégies peuvent être considérées :

- A. Nous faisons les calculs des temps de début et de fin au plus tard avec les durées des tâches mères en faisant des calculs sur les pires cas et meilleurs cas de réalisation (en tenant compte aussi bien des durées des tâches qui ont forcément lieu que des tâches optionnelles).
- B. Nous faisons les calculs des temps de début et de fin au plus tard en tenant compte uniquement des durées des tâches qui ont forcément lieu. Nous mettons à jour les calculs en tenant compte des durées des tâches optionnelles si elles ont lieu concrètement.

SIMULATION D'UN SCÉNARIO

Le simulateur

Pour valider notre approche, nous avons développé un simulateur pour l'assistance à l'utilisateur et le suivi de ses tâches dans les environnements ambiants. Ce simulateur prend en entrée le fichier issu de la conception d'un modèle de tâche à l'aide l'éditeur CTTE [16]. Cet outil peut donc être utilisé par le concepteur pour saisir son modèle de tâche comme illustré sur la figure 9. Le modèle résultant est intégré directement dans notre simulateur.

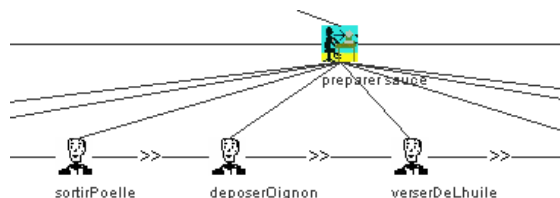


Figure 9. Exemple d'une partie d'un arbre saisi avec CTTE

Pour chaque tâche, un ensemble d'informations est saisi (son identifiant, sa fréquence, ses durées maximum et minimum de réalisation,...). En ce qui concerne l'ensemble de propriétés qui ne sont pas prises en compte dans le modèle CTT (les services appelés, le temps maximum d'attente d'activation et l'ensemble de ses pré et post-conditions), le concepteur peut utiliser le champ de texte

« description » dans CTTE. Le simulateur analyse le fichier en entrée généré par CTTE. Pour chaque niveau de l'arbre de tâches, cet analyseur conserve les opérateurs temporels et restructure l'arbre de tâches en se référant aux priorités de ces opérateurs de manière à produire un arbre binaire sans liens horizontaux.

La figure 10 montre un exemple du modèle de tâche résultat sous forme d'un arbre binaire au sein du simulateur. Ce dernier permet de visualiser les opérateurs temporels reliant les différentes tâches concrètes ainsi que les informations les plus pertinentes concernant ces tâches et pouvant être utiles lors du déroulement de la simulation :

- le nom de la tâche ou son identifiant
- son état (calculé au moment de l'exécution)
- les informations relatives à ses pré-conditions :
 - la liste de ses pré-conditions et leurs états que le concepteur peut modifier au moment de la simulation.
 - l'urgence de l'activation de la tâche en fonction du délai restant par rapport au temps maximum d'attente d'activation qui lui a été prescrit au moment de la conception. Le voyant vert de la rubrique « Pré-conditions » indiquera qu'il reste encore du temps pour la réalisation des pré-conditions et le voyant rouge s'allumera pour indiquer la nécessité de les satisfaire de manière urgente.
- les informations relatives à sa réalisation :
 - l'urgence de la réalisation de la tâche en fonction des alertes reçues de la part du contrôleur de tâches abstraites CTA. Le voyant vert de la rubrique « Réalisation » indique qu'il reste encore du temps pour la réalisation de la tâche. Le voyant orange s'allume pour indiquer qu'il ne reste pas assez de temps pour la réalisation des tâches suivantes en s'appuyant sur une projection pessimiste. Enfin le voyant rouge indique que le temps restant ne sera pas suffisant pour la réalisation des tâches restantes même en faisant une projection optimiste.
 - la barre de progression se remplit en fonction du temps écoulé lors de la réalisation de la tâche. La couleur de remplissage est bleue si on n'a pas encore dépassé la durée de réalisation qui a été prévue et rouge dans le cas contraire.
 - un bouton « Start/ End » permet de simuler la détection du démarrage d'une tâche et celle de sa fin de réalisation.

Déroulement du scénario

Le simulateur que nous avons développé permet au concepteur de simuler la production d'informations provenant des capteurs de l'environnement. Pour la tâche « SortirPoêle » les pré-conditions qui lui ont été assignées sont « placardOuvert » et « PoêleDansLePlacard » et

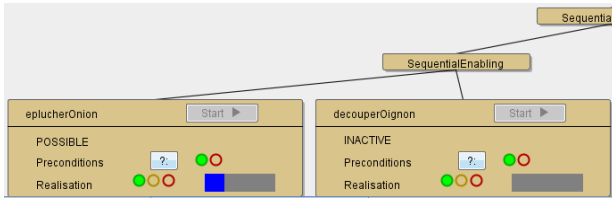


Figure 10. Notre arbre de tâches binaire

sont reliées par un « ET » logique. Le concepteur peut simuler la satisfaction des pré-conditions en cochant les deux cases appropriées du simulateur. L'expression logique associée est alors évaluée par le simulateur et la faisabilité de cette tâche est donc calculée en runtime et elle devient réalisable comme sur la figure 11. Il devient alors possible de cliquer sur bouton « Start » relatif à la tâche pour simuler la détection de son démarrage.

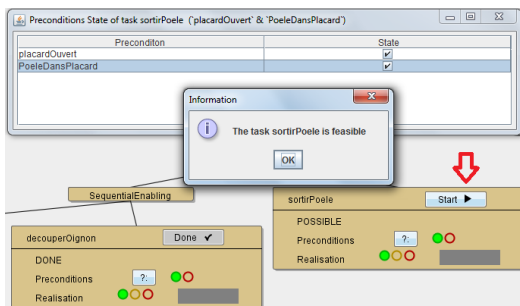


Figure 11. Vérification des pré-conditions d'une tâche qui devient faisable

Supposons que l'utilisateur essaye de commencer la préparation des pâtes alors qu'il n'a pas encore mis en marche la gazinière. Le rôle du CTC peut être illustré par une notification sur le fait que la tâche est non réalisable comme le montre la figure 12.

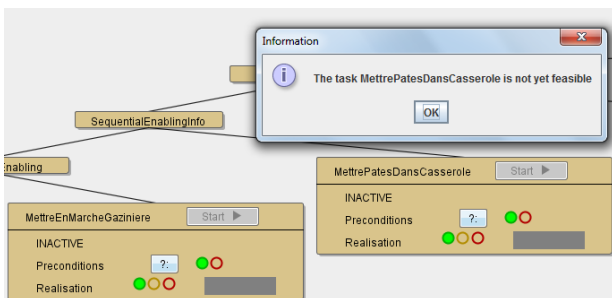


Figure 12. le CTC notifie une tâche non réalisable voulant démarrer

Afin d'illustrer le rôle du CTA, considérons la partie suivante du scénario : dès que l'utilisateur décroche le téléphone pour répondre à un appel sa tâche courante « éplucher oignon » est suspendue. Pendant ce temps, le temps de réalisation de la tâche suspendue continue tout de même à avancer comme l'indique la figure 13.

Au moment où l'utilisateur finit de répondre à l'appel téléphonique sa tâche suspendue « éplucher oignon » peut être reprise. Comme le temps maximal prévu a été dépassé, la barre de progression est affichée en rouge sur la figure 14. De plus il ne restera plus de temps pour finir le

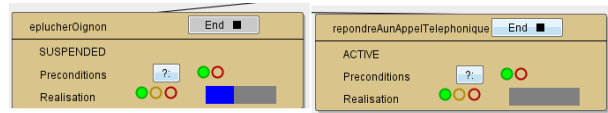


Figure 13. L'activité en cours suspendue par un appel

reste des tâches même en les effectuant aux performances maximales (avec les temps minimums prédéfinis) et donc le voyant rouge de réalisation s'allume comme indiqué sur la figure 14.

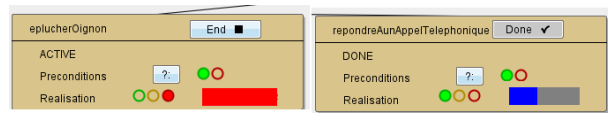


Figure 14. Reprise de la tâche suspendue suite à la fin de l'appel et notification par le CTA du retard engendré

CONCLUSION

Dans cet article, nous avons présenté un modèle de tâches adapté aux environnements ambiants, dont l'état peut être mis à jour au moment de l'exécution et qui étend la notation et la sémantique du modèle CTT (ConcreteTaskTree). Notre modèle permet de gérer dynamiquement l'état de chaque tâche et permet d'attribuer des états aux tâches au moment de l'exécution en fonction des informations échangées avec l'environnement (démarrage d'une tâche, fin de réalisation d'une tâche, états des pré-conditions...).

Nous avons également montré comment un système de suivi et d'assistance peut exploiter un tel modèle de tâches dynamique pour suivre au moment de l'exécution le déroulement des tâches, en fonction des états contenus dans le modèle de tâches et de ce qui se passe réellement dans l'environnement via les informations fournies par les capteurs. Nous avons détaillé ensuite le fonctionnement du système de supervision qui intervient afin d'alerter l'utilisateur.

Enfin nous avons présenté une validation de notre système à travers le déroulement d'un scénario sur notre simulateur. Cette simulation montre comment les interactions avec le modèle de tâches à l'exécution nous permettent de produire un système dynamique, qui prend en considération l'activité de l'utilisateur et lui fournit une aide pour la réalisation de ses tâches quotidiennes.

La prochaine étape importante de ce travail consistera à mener une évaluation expérimentale réelle, afin de pouvoir tester l'apport d'un tel système auprès d'utilisateurs finaux. Cette expérimentation nous permettra de déterminer les meilleures stratégies d'interaction avec l'utilisateur afin de trouver le bon équilibre entre un système d'assistance trop discret et un système trop intrusif. Il sera intéressant d'étudier l'application de l'approche à des domaines autres que l'informatique ambiante.

BIBLIOGRAPHIE

1. Annett J. Hierarchical task analysis. *Handbook of Task Analysis for Human Computer Interaction* (2004), 67-83.
2. Annett J. & Duncan K. Task analysis and training design. *Occupational Psychology* 41 (1967), 211-227.

3. Annett J., Duncan K. D., Stammers R. B. & Gray M. J. Task analysis. *London: Her majesty's stationery office.* (1971).
4. Balbo S., Ozkan N. & Paris C. Choosing the right task-modeling notation a taxonomy. In D. Diaper and S. Neville (Eds.), *The Handbook of Task Analysis for Human Computer Interaction* (2004), 445–465.
5. Baron M., Lucquiaud V., Autard D. & Scapin D. K-made : un environnement pour le noyau du modele de description de l'activite. *18eme Conference Francophone sur l'Interaction Homme-Machine (IHM'2006)* (2006), 287–288.
6. Ducatel K., Bogdanowicz M., Scapolo F., Leijten J. & Burgelman J.-C. Scenarios for ambient intelligence in 2010. *Final report, Information Society Technologies Advisory Group (ISTAG)* (2001).
7. Gharsellaoui A., Bellik Y. & Jacquet C. Requirements of task modeling in ambient intelligent environments. *International Conference on Ambient Computing, Applications, Services and Technologies* (2012), 71–78.
8. Giersich M., Forbrig P., Fuchs G., Kirste T., Reichart D. & Schumann H. Towards an integrated approach for task modeling and human behavior recognition. *HCI 2007 4550* (2007), 1109–1118.
9. Johnson P. Human-computer interaction: Psychology, task analysis and software engineering. *London: McGraw-Hill* (1992).
10. Johnson P., Diaper D. & Long J. Tasks, skill and knowledge: Task analysis for knowledge based descriptions. In *Proceedings of First IFIP Conference on Human-Computer Interaction (Interact '84) 1* (1984), 23–28.
11. Kindberg T. & Fox A. System software for ubiquitous computing. *IEEE Pervasive Computing 1* (2002), 70–81.
12. Klug T. & Kangasharju J. Executable task models. *TAMODIA 2005* (2005), 119–122.
13. Limbourg Q. & Vanderdonck J. Comparing task models for user interface design. *The Handbook of Task Analysis for Human-Computer Interaction* (2003), 135–154.
14. Limbourg Q., Vanderdonck J., Michotte B., Bouillon L. & Lopez-Jaquero V. Usixml: A language supporting multi-path development of user interfaces. *Engineering Human Computer Interaction and Interactive Systems 3425* (2005), 200–220.
15. Mahfoudhi A., Abed M. & Tabary D. From the formal specifications of user tasks to the automatic generation of the hci specifications. *People and Computers XV* (2001), 331–347.
16. Mori G., Paterno F. & Santoro C. Ctte: Support for developing and analyzing task models for interactive system design. *IEEE transactions on software engineering 28* (2002), 797–813.
17. Ormerod T. C., Richardson J. & Shepherd A. Enhancing the usability of a task analysis method: a notation and environment for requirements specification. *Ergonomics, 41(11)* (1998), 1642–1663.
18. Ormerod T. C. & Shepherd A. Using task analysis for information requirements specification: The sub-goal template (sgt) method. *The Handbook of Task Analysis for Human-Computer Interaction* (2004), 347–365.
19. Ozkan N., Paris C. & Balbo S. Understanding a task model: An experiment. *Proceedings of the HCI on People and Computers XIII* (1998).
20. Paterno F. Model based design and evaluation of interactive applications. *Berlin: Springer-Verlag* (1999).
21. Paterno F. Task models in interactive software systems. *Handbook of Software Engineering and Knowledge Engineering 1* (2002), 1–19.
22. Patrick J. Training : research and practice. *London: Academic Press.* (1992).
23. Patrick J., Gregov A. & Halliday P. Analysing and training task analysis. *Instructional science* (2000), 51–57.
24. Riva G. The psychology of ambient intelligence: Activity, situation and presence. *Ambient Intelligence: The Evolution of Technology, Communication and Cognition Towards the Future of Human-Computer Interaction (Emerging Communication)* (2005), 17–33.
25. Riva G., Loreti P., Vatalaro M., Vatalaro F. & Davide F. Presence 2010: The emergence of ambient intelligence. *Being There: Concepts, effects and measurement of user presence in synthetic environments* (2003), 60–81.
26. Sadri F. Ambient intelligence: A survey. *ACM: Computer Survey 43, 4* (2011), 36–66.
27. Schulungbaum E. Model-based user interface software tools current state of declarative models. *Visualization and Usability Center* (1996).
28. Sinnig D., Chalin P. & Khendek F. Consistency between task models and use cases. *Proceedings DSV-IS 2007* (2007).
29. Sinnig D., Wurdel M., Forbrig P., Chalin P. & Khendek F. Practical extensions for task models. *TaMoDia 2007* (2007), 42–55.
30. Siochi A. & Hartson H. Task-oriented representation of asynchronous user interfaces. *CHI '89 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1989), 183–188.
31. Tapia E. M., Intille S. S. & Larson K. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing* (2004), 158–175.
32. Tarby J.-C. & Barthet M.-F. The diane+ method. *Computer-aided design of user interfaces* (1996), 95–120.
33. VanderVeer G. C., VanderLenting B. F. & Bergevoet B. A. J. Gta: Groupware task analysis - modeling complexity. *Acta Psychologica 91* (1996), 297–322.
34. Weiser M. The computer for the twenty-first century. *Scientific American* (1991), 94–100.
35. Weiser M. Some computer science issues in ubiquitous computing. *Communications of the ACM 36, 7* (1993), 75–84.
36. Wurdel M. & Forbrig P. A comparative study of task concepts in usixml and ctml and their applications. *First International workshop on User Interface eXtensible Markup Language (UsiXML'2010)* (2010), 173–182.
37. Wurdel M., Sinnig D. & Forbrig P. Towards a formal task-based specification framework for collaborative environments. *Computer-Aided Design of User Interfaces VI* (2009), 221–232.

Drag-and-drop for older adults using touchscreen devices: effects of screen sizes and interaction techniques on accuracy

Lilian Genaro Motti

IRIT UMR 5505, Université
de Toulouse 3 Paul Sabatier
118 Route de Narbonne
31062, Toulouse, France
genaro[at]irit.fr

Nadine Vigouroux

IRIT UMR 5505, Université
de Toulouse 3 Paul Sabatier
118 Route de Narbonne
31062, Toulouse, France
vigourou[at]irit.fr

Philippe Gorce

HandiBio EA 43 22, Université
du Sud Toulon Var
Avenue de l'université
83957, Toulon, France
gorce[at]univ-tln.fr

ABSTRACT

This study investigates the accuracy of drag-and-drop interaction for older adults by analyzing the number of supplementary attempts for positioning a target during the execution of tactile puzzle games on two different screen sizes, tablet and smartphone, with finger and pen interaction. 24 older subjects (aged 65 to 86) participated of the experiment. The results showed that there is a significant effect of the interaction techniques during interaction on smartphone. Subjects were more accurate during pen interaction on both screen sizes. Age effects were significant but subjects aged 80 years old or oldest sometimes performed better than subjects aged 70 to 79 years old, especially during pen interaction. This study shows that drag-and-drop is an efficient technique for moving targets even on small touchscreen devices and pen interaction may help older users to execute more accurate drag-and-drop interaction on touchscreen devices.

Keywords

Touchscreen, older adults/elderly, interaction techniques, drag-and-drop, pen interaction, serious game.

ACM Classification Keywords

H 5.2 User interfaces: Input devices and strategies (e.g. mouse, touchscreen)

INTRODUCTION

Touchscreen interaction has been recommended for older adults for several reasons: devices are less intimidating [18], direct interaction on the display requires less cognitive, spatial or attentional demand [21] so older users performances are better than compared with others input devices [2] Touchscrees also reduces the age-related differences of interaction when compared to traditional input devices [3,4,16].

There is still some reluctance of older adults on adopting new technologies and usability is one of the reasons [23].

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670460>

Improving interaction by supporting users and preventing errors is an important matter for older adults because errors usually cause supplementary gestures, increasing the motor and cognitive load for accomplishing a task.

Serious games can facilitate the discovering of new technologies and interaction techniques by older adults [14]. The system Puzzle Touch has been designed to help older users to discover touchscreen technologies and learn tactile interaction. To the present study, this system has been adapted to evaluate drag-and-drop gestures of interaction by tracking every touch on the screen. This study investigates the effects of the screen size and interaction techniques for moving targets during the execution of tactile puzzle games. 24 volunteers aged 65 to 86 years old executed a series of puzzle games on two different devices, a smartphone and a tablet, with pen or finger interaction. The main task consisted of moving 12 pieces randomly placed on the bottom part of the screen and positioning them at their correct emplacement on a grid displayed on the top of the screen in order to recompose the original image.

The present study aims to further expand the understanding of the tactile interaction of older users. For this analysis, we will study the number of supplementary attempts for positioning a target (a piece of the puzzle game) according to the interaction technique and the two accuracy levels required on each screen size. The design of the system allows two levels of difficulty for positioning the targets.

The next section present some related work about the evaluation of older users interaction on touchscreen and drag-and-drop interaction. Then, the Puzzle Touch system is described on section 3, followed by the experiment procedures on section 4. Results are presented on section 5. Section 6 presents a discussion and future work. Finally, conclusion is presented on section 7.

RELATED WORK

Previous studies about touchscreen interaction by older adults evaluate different interaction gestures. Single and multi-touch gestures, including tapping, drag-and-drop,

pinching and steering are commonly used for target selection, typing tasks and manipulating objects on touchscreen.

Kobayashi et al (2011) [10] observed that tapping was easy to learn for novice older users and a week of use increased pinching and dragging performances. They reported user's preferences for dragging and pinching rather than tapping. Dragging performances also increased after a week practice. Findlater et al (2013) [4] compared different gestures on touchscreen and a mouse. They found that dragging was the slowest on the touchscreen. Stöbel et al (2009) [17] evaluated the execution of patterns of gestures on touchscreen devices. They reported that older users preferred more accurate gestures even if it takes longer times. The distance of the trajectories is related to the screen sizes, requiring more amplitude of movements. This effect has been reported as causing deviation during the movements [17] but the accomplishment of dragging tasks is reported to be faster on bigger screen sizes [10].

Drag-and-drop interaction on touchscreen has been proposed to motor impaired users [5] and older users with tremor [12,20] because the continuous contact with the screen have an effect on finger oscillation, helping users to increase the accuracy. It has also been used with cognitive impaired people and mouse interaction [19]. As well as swabbing, drag-and-drop has been used for target acquisition, digit and text entry tasks with pen [5] and fingers [12,20].

During dragging tasks, slower movements can be related to the fact that the hand occludes a part of the screen [11] or the perception of the necessary friction to move the target. Pen interaction could be used to solve this problem but only few studies evaluated pen interaction for older users. Moffat & McGrenere (2007) [13] investigated the causes of target selection difficulties during pen interaction and observed that slipping is a common error for older adults, while drifting or missing just below occur to all age groups. This kind of errors also affects typing tasks, provoking omission, substitution or insertion of characters [22]. Nicolau & Jorge (2012) [15] have observed that typing errors are correlated to hand tremor and suggested bigger targets to help older users.

Few studies report the supplementary manipulation an user has to do to recover from an error [1] or other supplementary gestures. Harada et al (2013) [7] have observed that novice users make shorter gestures for panning or scrolling tasks.

The number of supplementary gestures could be a relevant criterion for evaluating interaction. Missing a target can enchain supplementary manipulations to accomplish an action, which can be troublesome for older users because supplementary gestures can trigger other errors. Sometimes, the user needs to start again from the beginning [1]. For the present study, the

supplementary gestures for correcting the positioning of a target will be used as an evaluation criterion for accuracy.

THE PUZZLE TOUCH SYSTEM

The Puzzle Touch System is a game destined to help older users discovering of touchscreen technologies and learning tactile interaction gestures. This is the third phase of a user-centered design method for improving this system towards a serious game. This version of the system applies the results of previous studies and the parameters of the game have been chosen to allow the evaluation of drag-and-drop interaction on two screen sizes with two interaction techniques and two accuracy levels.

Images from old colored postcards provided by the city hall archives have been used to generate puzzle games. These images present different views and times of the city subjects live in, helping to arouse their interest and create social interaction [14]. During the demonstration meetings, users can play in groups and share their experience about technologies. When playing games on touchscreen devices, users engage to two personal challenges: learning an interaction gesture and solving the puzzle (Figure 1).

At the beginning of the game, 12 rectangular pieces are placed at the bottom of the screen and should be moved by pen or finger interaction to their corresponding emplacement on a 3x4 grid, presented on the mid-top of the screen (Figure 2). Images have been homogenized on amount of colors and contrast levels. A watermark placed as a background for the grid provides support for the user, reducing the cognitive workload by providing visual cues.

When the interaction technique (pen or a finger) touches a piece, it comes on the top of the others. The piece follows the movement of the pen or the fingertip (**drag**). When the pen or the finger leaves the screen, the piece stops moving (**drop**) and the system test the piece's emplacement. If the piece is placed on its correct emplacement, a visual feedback is shown (a "flash") and the piece can no longer be moved. If the user touches a placed piece, the "flash" effect appears as a confirmation

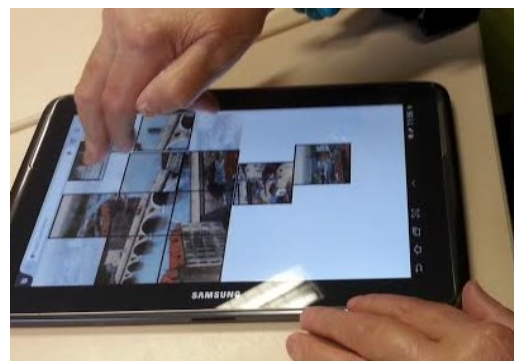


Figure 1 Older adult playing a tactile puzzle on a tablet



Figure 2 A screenshot of the system Puzzle Touch on a smartphone

of its validation. When the 12 pieces are correctly placed on the grid, a congratulation message is displayed and the experimenter selects another game.

The puzzle games were built on HTML5, CSS3, Javascript and Php and available on two touchscreen devices through Mozilla Firefox mobile web browser. Targets sizes (a puzzle piece) are 19x19mm on the smartphone (85 pixels width) and 35x35mm (195 pixels width) on the tablet. In order to compensate the lack of spacing between the targets, they are contoured by a 1mm dark border.

The system starts counting the number of attempts once a piece is already placed on its right emplacement. We have considered the right emplacement when the piece covers 50% of its emplacement on the grid (Figure 3). The number of attempts represents the supplementary gestures for correcting the positioning of the target. The positioning is verified when the user leaves the finger or pen according to the required accuracy levels. Normally, 12 precise gestures are needed to complete the game.

The SNA represents the Supplementary Number of Attempts for reaching the target (correctly positioning a puzzle piece on the grid) according to the required accuracy levels. The system was designed with two accuracy levels to validate the correct emplacement of a piece: 80% of covering for lower accuracy level and 95% for higher accuracy level. There are two levels on the smartphone and two levels on the tablet. These covering requirements have been proportionally adapted (ratio= 2.29) to the targets sizes on each screen size.



Figure 3 Example of positioning: A) the piece is covering 60% of its emplacement and should be correctly positioned, B) the piece is covering 100% of its emplacement and the positioning is validated

Blank tests with old adults and demonstrations meetings validated the different levels of accuracy, our design choices and the playability of the game.

The system allows multi-touch interaction; two or more pieces can be moved at the same time. Accidental touches outside the game zone have not been blocked (buttons, menus and tactile shortcuts). The game can be restarted if necessary.

METHODOLOGY

The following analysis study the SNA according to the different conditions for executing the puzzle game: two accuracy levels, finger and pen interaction on two different screen sizes, a tablet and a smartphone. We will look for the effects of interaction technique, accuracy requirements and age on the different screen sizes.

Apparatus

The chosen devices have different screen sizes: a 5.5 inches screen smartphone Galaxy Note II (WXGA 1280x720 Super AMOLED) and a 10.1 inches screen tablet Galaxy Note 10.1 (WXGA 1280x800 LCD). Both allow interaction with pen and finger.

Recruitment

The Puzzle Touch system was presented on associations, senior clubs and libraries where older adults were used to frequent or take computer lessons at Toulouse. Being aged of 65 years old or more was the only criteria of inclusion. After a demonstration meeting, volunteers had an appointment for an individual session. Participants signed a written consent form.

Procedure

The activity took place in a quiet place (separated room or library) with artificial light at the sealing. All users were seat and placed the devices horizontally on the table. They were told to install themselves comfortably and to execute the movements with accuracy.

Every subject passed a learning phase (at least 4 practice games with both devices and interaction techniques at the lower accuracy level).

Then they passed sight tests. The goal of this pre-experiment measurement is to provide information about visual acuity or impairment of participants to better understand user's difficulties during interaction. Three applications for eyesight tests, provided by healthcare4mobile¹, installed and displayed at the 5.5 inches screen of the smartphone, were chosen to measure color perception, central vision acuity and contrast sensitivity. The screen was held vertically about 30 cm from their faces. Subjects who wore glasses were told to keep them. The first test was designed to identify color deficiency. It presents 6 colored images and subjects should identify the number inside the pattern. The second test was designed to reveal visual impairment on a person's central visual field (which could reveal age-related macular degeneration). Subjects should cover one eye and describe two images displayed on the screen, presenting a small dot in the center of a grid. They repeated the procedure covering the other eye. The third was designed to test contrast sensitivity. Eight images presenting numbers were displayed on the screen and subjects should read them, covering one eye then the other. The size and the contrast of the numbers reduced from one image to the following. At the end, participants respond to the third test with eyes uncovered.

Then subjects were questioned about any motor control problem, injury or difficulty of accuracy that could affect hands, fingers, arms or upper-limb movements. They were not questioned about their cognitive capabilities.

After, subjects were questioned about their previous experience with technologies: how often they use computers, cell phones, tablets and smartphones, and if they have any of these devices.

Finally, they played 8 games at total, 4 games on each device (smartphone and tablet): with 2 interaction techniques (pen and finger) and 2 accuracy levels (80% and 95%). Therefore 8 images were selected generating one puzzle game for each condition. Devices, interaction techniques and accuracy levels have been counter balanced.

Measures

In order to analyze the accuracy of the gestures of interaction, we will focus on SNA criterion.

For the analysis of the interaction, we will use three independent variables:

- Accuracy level: 80% covering (lower accuracy) or 95% covering (higher accuracy).
- Interaction technique: pen or fingers.
- Age range.

¹ <https://play.google.com/store/apps/developer?id=healthcare4mobile>

For the analysis of the user's characteristics, we collected information about:

- Age
- Visual impairment
- Motor impairment
- Previous experience and frequency of use of technologies: computer, cell phone, tablet and smartphone.

Additionally, the experimenter filled up an observation grid and took notes about the interaction.

When the user touches the game zone outside the pieces, there isn't any action of the system but it registers the coordinates of the touch in order to allow further investigation about accidental touches. The targets positions, the movement's paths and the drops of the pieces have been stored for future analyses of strategy and errors. Only the tracks of full games have been registered and used for this analysis. All data have been anonymized and kept confidential on a secure server.

Participants

24 older adults aged of 65 years or more (65-86, mean 74.25, SD 5.8), 8 men (mean 73.12, SD 5.66) and 16 women (mean 74.81, SD 5.96), participated of the experiment. The number of subjects on each age range is represented on Figure 4.

All of them had visual impairments. 3 of them didn't wear glasses. 2 had no or little sight on one eye due to injuries but sufficient correction for the healthy eye. Sight control before the experiment shows that none of them had problems to distinguish colors or a grid displayed on the screen placed near them. On the other hand, contrast sensitivity test shows that 2 persons had insufficient sight correction.

They were all right handed, 3 women were ambidextrous but used the right hand to interact. 3 women had dexterity problems and 1 had lost sensitivity due to injuries on one or more fingers of the dominant hand. 3 women had arthrosis on the hands. 2 subjects had tremor (1 woman, 1 man).

All the volunteers were interested on technologies, their impact on our daily lives, their usability and usefulness, as well as security and privacy issues. Most of them took computer lessons and felt concerned by its development.

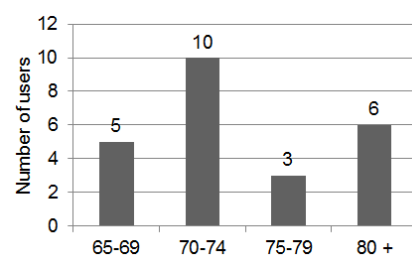


Figure 4 Number of subjects on each age range

16 participants used computers regularly and 8 used it rarely or never. 15 participants used mobile phones regularly; the others 9 used it rarely or never. 5 of them had a tactile mobile phone; the others had never used a smartphone before.

5 participants had a tablet, 3 of them used it every day or almost every day. 2 of them used it occasionally because they had difficulties using it. 18 subjects had never used a tablet before.

RESULTS

Observation results

All the subjects used the right hand to interact. Index and major were the common used fingers for finger interaction. Thumb, index and major were the three fingers used to hold the pen during pen interaction. No subject tried multi-touch interaction; they moved one piece at a time.

Statistical analysis of SNA

For a statistical analysis data normality was checked with Shapiro-Wilk tests. Results returned negative for the NSA data series on smartphone (W= 0.8307, p-value= 4.149e-09) and on tablet (W= 0.6963, p-value= 8.586e-13). By consequence, data were analyzed with the non-parametric analysis of variance Friedman test.

The following diagrams describe SNA for all subjects on each condition on smartphone (Figure 5) and on tablet (Figure 6).

Interaction on smartphone

Friedman test showed statistically significant differences on SNA between the different conditions of the experiment on smartphone (chi-squared= 57.0639, df= 23, p-value= 0.0001003).

A post-hoc analysis with Wilcoxon Signed Rank Tests was conducted with a Bonferroni correction applied, resulting in a significance level set at p<0.0125.

Results of Wilcoxon Signed Rank Tests showed a significant difference for interaction techniques (Z= 0.5384686, p-value= 0.006565) and significant difference for accuracy levels (Z= -6.030849, p-value= 2.457e-09).

Table 1 shows mean SNA for all subjects for pen and finger interaction on smartphone during each accuracy level.

SNA (Smartphone)	Pen	Finger
Low accuracy level (80%)	2.04 (SD=3.03)	4.50 (SD= 4.98)
High accuracy level (95%)	22.04 (SD= 12.41)	26.13 (SD= 16.40)

Table 1 Mean SNA on smartphone for each accuracy level

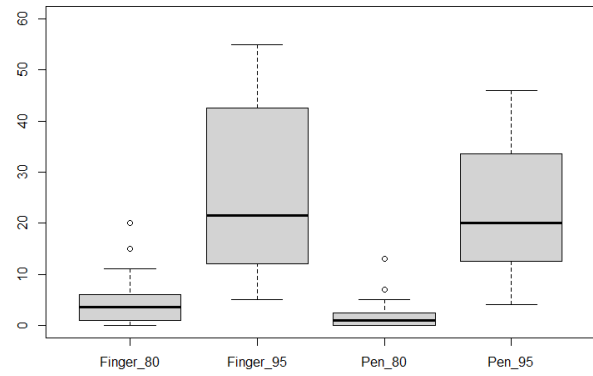


Figure 5 SNA during interaction on smartphone with finger or pen on low (80%) or high (95%) accuracy levels

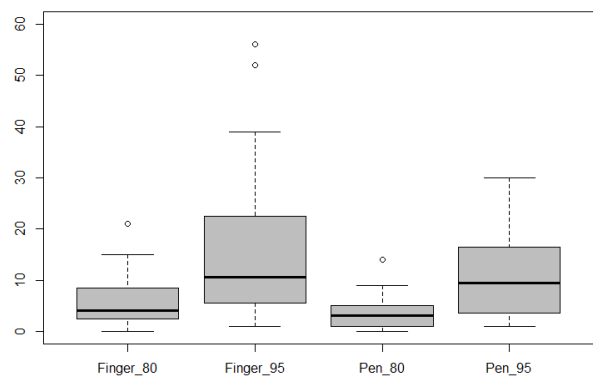


Figure 6 SNA during interaction on tablet with finger or pen on low (80%) or high (95%) accuracy levels

The mean SNA for all the 24 subjects showed that pen interaction was more accurate than finger interaction for both accuracy levels on smartphone.

Interaction on tablet

Friedman test showed statistically significant differences in SNA between the different conditions of the experiment on smartphone (chi-squared= 56.9907, df= 23, p-value= 0.0001028).

A post-hoc analysis with Wilcoxon Signed Rank Tests was conducted with a Bonferroni correction applied, resulting in a significance level set at p<0.0125.

Results of Wilcoxon Signed Rank Tests showed no significant difference for interaction techniques (Z= 0.394877, p-value= 0.02874) and significant difference for accuracy levels (Z= -5.625715, p-value= 2.852e-08).

Table 2 shows mean SNA for all subjects for pen and finger interaction on tablet during each accuracy level.

Even if there was no significant difference for interaction techniques during tablet interaction, the mean SNA for all subjects on tablet showed that pen interaction was more accurate than finger interaction on both accuracy levels.

SNA (Tablet)	Pen	Finger
Low accuracy level (80%)	3.50 (SD= 3.40)	6.00 (SD=4.98)
High accuracy level (95%)	12.58 (SD= 13.30)	18.04 (SD=16.95)

Table 2 Mean SNA on tablet for each accuracy level

The statistical analysis showed a big variability of SNA for the higher accuracy level for both interaction techniques. This variability is more important on smartphone.

For better understanding this variability among the participants, we looked for potential age effects.

Age effects

The collected data was divided according to the different age ranges. Results of Friedman tests show significant effect of age (chi-squared= 16.35, df= 3, p-value= 0.0009612).

Figure 7 and Figure 8 describe SNA during interaction on smartphone and tablet, respectively, according to the age range.

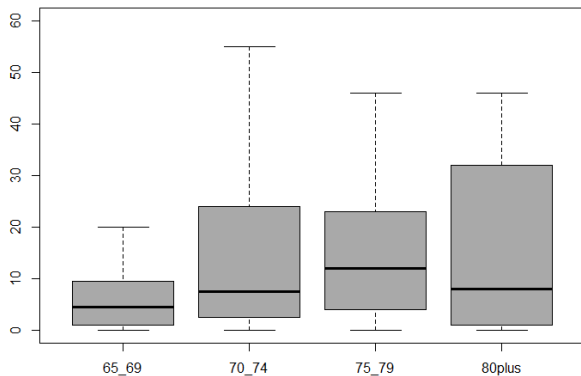


Figure 7 SNA during interaction on smartphone according to the age range

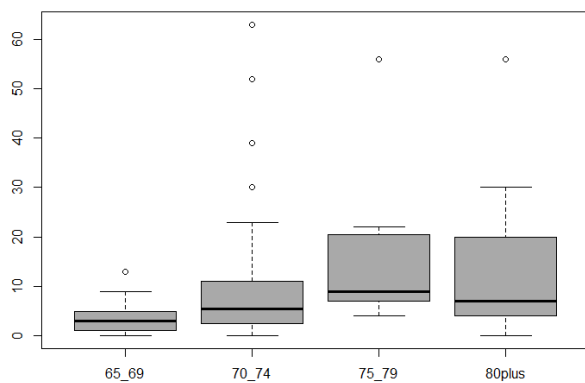


Figure 8 SNA during interaction on tablet according to the age range

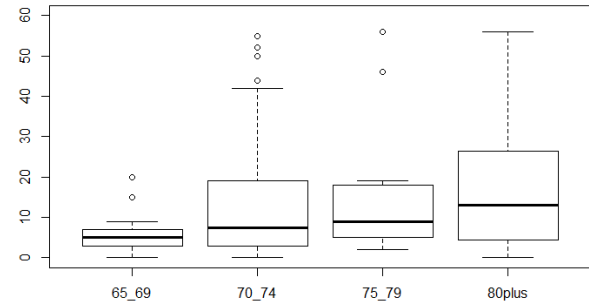


Figure 9 SNA during finger interaction according to the age range

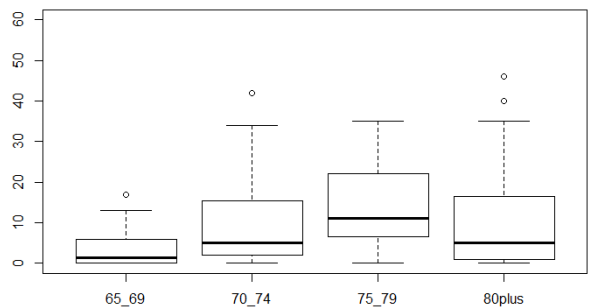


Figure 10 SNA during pen interaction according to the age range

Because of the aging effects on sensorial and cognitive skills [2] it would be expected that younger subjects interact more accurately, executing less SNA.

However, when analyzing each condition separately, subjects aged 80 years old or older sometimes performed better than younger subjects.

For better understanding these results, potential interaction techniques effects have been searched.

Results of the Wilcoxon Signed Rank test for subjects aged 80 years old or older showed significant effects for interaction techniques (Z= 1.528571, p-value= 0.01295). However, there were no significant effects for the interaction techniques for the other age ranges.

Figure 9 shows SNA during finger interaction and Figure 10 shows SNA during pen interaction according to the age groups. Subjects aged 80 years old or more were more accurate during pen interaction on both accuracy levels: mean SNA for 80 years old or older 11.08 (SD= 13.87), for subjects aged 75 to 79 14.17 (SD= 19.69), for subjects aged 70 to 74 11.23 (SD= 13.44) and for subjects aged 65 to 69 3.95 (SD= 5.21).

DISCUSSION

Tactile puzzle games for discovering touchscreen devices and learn tactile interaction

The Puzzle Touch system allowed novices and experienced older users to learn and accomplish tactile interaction during the execution of puzzle games on touchscreen devices.

During the demonstration meetings, this game helped older adults to discover touchscreen devices and drag-and-drop interaction with pen and fingers.

It is important to highlight that most of the participants of this study were interested on information and communication technologies (ICTs), 16 of them using a computer very often and 15 of them using cell phones. There is still some reluctance of older populations on adopting and using technologies. Games and social activities can help to change their negative subjective evaluation of technologies.

Besides, in addition to perceived usefulness, usability is an important issue for improving ICT's acceptance, as pointed out by Hanson (2012) [6] and Zhou and al (2012) [23]. Including older users as participants during the design phases of an interactive system or device is important to improve usability of touchscreen devices preventing digital exclusion.

SNA as evaluation criterion

The statistical analysis of SNA on the present study allowed the assessment of the accuracy of the interaction gestures of older adult using touchscreen.

The number of supplementary gestures is an important issue of usability for older users and the results of this study show that this criterion can be used for qualifying interaction.

We propose that the SNA should be associated to other criteria for studies of human-computer interaction, complementary to time, accuracy rates and subjective assessment.

Screen size and interaction technique effects

The results of the statistical analysis of SNA showed a significant effect of the interaction techniques during interaction on smartphone. Subjects had less SNA during pen interaction, so they were more accurate.

Hourcade & Berkel (2006) [8] evaluated pen or finger interaction for target selection tasks with tapping gestures on a small screen device (about 3.5). Their results showed finger interaction were more accurate than pen interaction on time and accuracy. But older users were capable to perform 87.6 % accuracy rates during pen interaction tapping 16mm targets. Our results show that pen interaction may help older users to execute more accurate drag-and-drop gestures on touchscreen devices. Even if there was no significant difference for interaction techniques during tablet interaction, subjects also had less SNA during pen interaction on larger screen size and on both requirements for accuracy.

Several authors suggest bigger targets to increase accuracy rates and they highlight the importance of preventing errors [8,9,10,13,15]. Our results also show less SNA on bigger screen size, where targets were bigger. Lowering the accuracy requirement could facilitate interaction with small targets. Adapting the systems requirements for accuracy according to the

screen sizes could also be helpful for target selection or even text entry tasks.

Age effects and user's profile

Age effects were significant, but the analysis of SNA showed that the oldest group of users sometimes performed better than others groups. Leonard et al (2005) [21] evaluated drag-and-drop during pen interaction for older users with visual impairment. Subjects were able to interact with accuracy but the severity of the visual impairment was a predictor for interaction efficiency. A more significant number of participants would help us to understand the effects of user's profile, including sensorial, cognitive or motor impairments.

We have found significant effects of the interaction techniques according to the age groups especially for the oldest group (80 or older) that were more accurate during pen interaction. Further studies should be done to identify the effects of user's background as previous experience with computers, technologies and touchscreen devices on interaction performances.

Future work

Future work should compare SNA to other evaluation criteria as time and accuracy rates. Comparing performances of older and younger populations should provide information to better understanding the variability on SNA and help to characterize the older population.

More numbers of attempts could also indicate different strategies of interaction. The analysis of the trajectories could provide cues for developing new interaction gestures.

Further studies should evaluate the effects of drag-and-drop on common errors related to older users' tactile interactions such as slipping errors in order to improve interaction.

CONCLUSION

This study has analyzed the accuracy of drag-and-drop during the execution of tactile puzzle games on touchscreen devices with finger and pen interaction on two different screen sizes, smartphone and tablet. The used criterion for evaluation was the number of attempts for positioning a puzzle piece (SNA).

Drag-and-drop was an efficient technique for moving targets even on small touchscreen devices. The results showed that there is a significant effect of the interaction technique during interaction on smartphone. Subjects were more accurate during pen interaction on both screen sizes and on both accuracy requirements.

Age effects were significant but subjects aged 80 years old or oldest sometimes performed better than subjects aged 70 to 79 years old, especially during pen interaction. Pen interaction may help older users to execute more accurate drag-and-drop interaction gestures on touchscreen devices.

Improving the usability of touchscreen devices would help older users to adopt and benefit of mobile devices, preventing digital exclusion.

ACKNOWLEDGMENTS

Phd Scholarship Ciências sem fronteiras, CNPQ, Brazil (#237079/2012-7).

We kindly thank all the participants and the organizations that helped us during the recruitment: Ateliers informatiques des Médiathèques Cabanis et Empalot and Association Générations Solidaires at Toulouse, France.

REFERENCES

- Bradley, M., Langdon, P., and Clarkson, P.J. Older User Errors in Handheld Touchscreen Devices : To What Extent Is Prediction Possible? C. Stephanidis (Ed.): *Universal Access in HCI, Part II, HCI 2011, LNCS 6766*, (2011), 131–139.
- Caprani, N., O'Connor, N., and Gurrin, C. Touch screens for the older user. In *Assistive Technologies, Fernando A. Auat Cheein (Ed.)*, ISBN: 978-953-51-0348-6, *InTech*, DOI: 10.5772/38302. 2012.
- Chung, M.K., Kim, D., Na, S., and Lee, D. Usability evaluation of numeric entry tasks on keypad type and age. *International Journal of Industrial Ergonomics* 40, 1 (2010), 97–105.
- Findlater, L., Froehlich, J.E., Fattal, K., Wobbrock, J.O., and Dastyar, T. Age-related differences in performance with touchscreens compared to traditional mouse input. *ACM CHI'13*, (2013), 343–346.
- Froehlich, J., Wobbrock, J., and Kane, S. Barrier pointing: using physical edges to assist target acquisition on mobile device touch screens. *ACM ASSETS'07*, (2007), 19–26.
- Hanson, V. Age and web access: the next generation. *18th International World Wide Web Conference WAA2009*, (2009), 7–15.
- Harada, S., Sato, D., Takagi, H., and Asakawa, C. Characteristics of Elderly User Behavior on Mobile Multi-touch Devices. P. Kotzé et al. (Eds.): *INTERACT 2013, Part IV, LNCS 8120*, (2013), 323–341.
- Hourcade, J.P. and Berkel, T. Tap or touch?: pen-based selection accuracy for the young and old. *ACM CHI'06*, (2006), 881–886.
- Ji, Y.G., Hwangbo, H., Yi, J.S., Rau, P.L.P., Fang, X., and Ling, C. The Influence of Cultural Differences on the Use of Social Network Services and the Formation of Social Capital. *International Journal of Human-Computer Interaction* 26, 11-12 (2010), 1100–1121.
- Kobayashi, M., Hiyama, A., and Miura, T. Elderly user evaluation of mobile touchscreen interactions. P. Campos et al. (Eds.): *INTERACT 2011, Part I, LNCS 6946*, (2011), 83–99.
- Lepicard, G. and Vigouroux, N. Comparison between single-touch and multi-touch interaction for older people. K. Miesenberger et al. (Eds.): *ICCHP 2012, Part I, LNCS 7382*, (2012), 658–665.
- Mertens, A. and Jochems, N. Design pattern TRABING: touchscreen-based input technique for people affected by intention tremor. *ACM SIGCHI EICS (2010)*, (2010), 267–272.
- Moffatt, K. and McGrenere, J. Slipping and drifting: using older users to uncover pen-based target acquisition difficulties. *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility - ASSETS'07*, (2007), 11–18.
- Motti, L.G., Vigouroux, N., and Gorce, P. Design of a social game for older adults using touchscreen devices and observations from an exploratory study. C. Stephanidis and M. Antona (Eds.): *UAHCI/HCI 2014, Part III, LNCS 8515*, (2014), 69–78.
- Nicolau, H. and Jorge, J. Elderly text-entry performance on touchscreens. *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '12*, (2012), 127.
- Schneider, N., Wilkes, Ja., Grandt, M., and Schlick, C. Investigation of Input Devices for the Age-differentiated Design of Human-Computer Interaction. *Human factors and Ergonomics Society 52nd annual meeting*, (2008), 144–148.
- Stöbel, C., Wandke, H., and Blessing, L. Gestural interfaces for elderly users: help or hindrance? S. Kopp and I. Wachsmuth (Eds.): *GW 2009, LNAI 5934*, (2010), 269–280.
- Umemuro, H. Lowering elderly Japanese users' resistance towards computers by using touchscreen technology. *Universal Access in the Information Society* 3, 3-4 (2004), 276–288.
- Vella, F., Vigouroux, N., and Rumeau, P. Investigating drag and drop techniques for older people with cognitive impairment. *Human-Computer Interaction. Users and ...*, (2011), 530–538.
- Wacharamanotham, C. Evaluating swabbing: a touchscreen input method for elderly users with tremor. *ACM CHI'11*, (2011), 623–626.
- Wood, E., Willoughby, T., Rushing, A., Bechtel, L., and Gilbert, J. Use of computer input devices by older adults. *Journal of Applied Gerontology* 24, 5 (2005), 419–438.
- Wright, P., Bartram, C., Rogers, N., and Emslie, H. Text entry on handheld computers by older users. *Ergonomics* 43:6, May 2013 (2000), 37–41.
- Zhou, J., Rau, P.-L.P., and Salvendy, G. Use and Design of Handheld Computers for Older Adults: A Review and Appraisal. *International Journal of Human-Computer Interaction* 28, 12 (2012), 799–826.

Direct and Indirect Multi-Touch Interaction on a Wall Display

Jérémy Gilliot¹, Géry Casiez² & Nicolas Roussel¹

¹Inria Lille, ²Université Lille 1, France

{jeremie.gilliot, nicolas.roussel}@inria.fr, gery.casiez@lifl.fr

ABSTRACT

Multi-touch wall displays allow to take advantage of co-located interaction (direct interaction) on very large surfaces. However interacting with content beyond arms' reach requires body movements, introducing fatigue and impacting performance. Interacting with distant content using a pointer can alleviate these problems but introduces legibility issues and loses the benefits of multi-touch interaction. We introduce WallPad, a widget designed to quickly access remote content on wall displays while addressing legibility issues and supporting direct multi-touch interaction. After briefly describing how we supported multi-touch interaction on a wall display, we present the WallPad widget and explain how it supports direct, indirect and de-localized direct interaction.

Key Words

Multi-touch; large display; direct interaction; indirect interaction; de-localized interaction.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation (e.g. HCI): User interfaces.

INTRODUCTION

Multi-touch wall displays make it possible for one or more people to interact with computing systems on an unprecedented scale. Co-localized touch-based interaction with the displayed content contributes to a high feeling of directness, but interaction with remote content can quickly become tiring and inefficient due to the required body movements. Objects of interest might be hard to perceive and reach if at the opposite end of the display, for example. Moving closer might help perceive them, but the objects might still remain out of arm's reach. Other users might also be standing on the way, or one might not want to move for some reason. These problems are even more acute when using legacy applications originally designed for smaller screen sizes: desktop applications heavily rely

on widgets that can not be moved easily (e.g. menus, toolbars and scrollbars) and that are automatically placed in locations far from ideal on large displays, for example.

Specific techniques are direly needed to properly support multi-touch interactions with legacy and modern applications on large displays. Previous work has investigated the combination of direct and indirect interaction [7], techniques to bring remote objects within arm's reach [2, 3] and interaction at a distance through an on-screen portal [10] or a mobile device [5]. However each of these solutions provides only partial answers to the above issues and multi-touch interaction at a distance in a multi-user context introduces additional challenges. In this paper, we present WallPad, a widget designed to address all of the above issues and the limitations of the current state of the art. WallPad supports elegant creation through a simple gesture, easy access to remote content, and precise direct, indirect and de-localized multi-touch interaction with it.

After a description of the context and our original motivation for this work, we present WallPad and its different features before presenting and discussing previous works related to multi-touch interaction on large displays.

CONTEXT AND MOTIVATION

Most Virtual Reality rooms built in the 1990s and 2000s were based on a large stereoscopic display. They typically used a 3D tracking system to support interaction with the virtual world, and a mouse and a keyboard to interact with the operating system and auxiliary 2D applications. At the time of their design, these VR rooms were clearly technologically advanced and considerable amounts of money were spent to realize them. But as interactive surfaces have become more and more popular in other contexts, the lack of support of their wall display for touch-based interaction certainly contributes to a diminished interest in them today. Our motivation for this work was the desire to upgrade an existing VR room to support multi-touch interactions.

Despite the many hardware and software technologies available for multi-touch interaction, supporting it in a VR room actually remains quite a challenge. The room configuration is often the result of different trade-offs and so cannot necessarily be easily changed. The screen is seldom flat and the image often produced by a carefully calibrated multi-projector system, for example. Ideally, one would like the hardware and software additions for multi-touch sensing to be cheap, easy to set up and with

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670445>

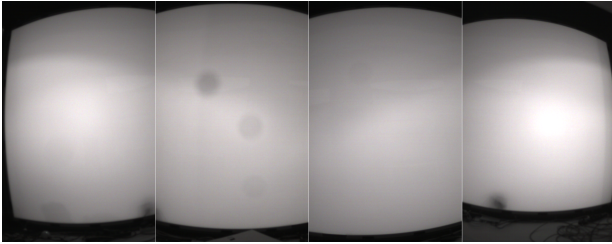


Figure 1. Raw images from our four cameras. The dark spots are caused by camera lens or filter stains.

a minimal impact on the existing uses of the room for VR applications, including the use of legacy 2D applications.

Initial room configuration

Inaugurated in 2006, the VR room in our research facility included about 400 k€ worth of equipment including a large display wall. The screen of this display is made of a $5.96 \text{ m} \times 2.43 \text{ m} \times 7 \text{ mm}$ acrylic layer constrained by a frame to follow a curved cylinder shape (76 cm sagitta) and coated on the front with a diffuser. Two specifically calibrated stereo projectors placed behind it make it possible to seamlessly display a composite image of 2392×1050 pixels.

To support multi-touch on such a large surface, optical sensing seemed the most affordable and practical solution. As the room had been literally built around the screen, changing the acrylic layer for an Endlighten¹ one or taking apart its frame to put LEDs on its side was not an option. Diffused surface illumination (DSI) and frustrated total internal reflection (FTIR) were thus quickly ruled out. The curved shape of the screen also ruled out laser and LED light planes (LLP and LED-LP). Rear diffuse illumination (DI) had been successfully used by the University of Groningen for their *Reality touchscreen*², with 1000 LEDs. We tried a similar approach by placing two illuminators taken from an Immersion iliGHT table behind our screen, but this did not work well. We had great difficulties achieving homogeneous rear IR illumination, the shininess of the acrylic layer causing numerous reflections and saturated areas.

During our tests, however, we observed that the room's lighting system emitted enough light in the IR spectrum to provide homogeneous front illumination. We thus decided to take advantage of this, acquired the appropriate hardware and started developing the necessary software to detect and analyze the IR shadows on the screen.

Hardware additions

The room's lighting system consists of 15 lamps attached to the ceiling plus 4 omni-directional ones on the walls and 4 others on the ground. All lamps are regular halogen bulbs which we set to their maximum intensity for maximum IR illumination.

¹<http://openmaterials.org/materials-101-light-diffusing-acrylic/>

²<http://www.rug.nl/science-and-society/centre-for-information-technology/hpcv/nieuws/touchscreen1>

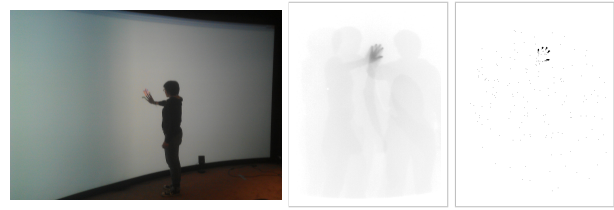


Figure 2. Background subtraction (middle) and adaptive thresholding (right) when touching the screen (left).

We use four IDS UI-1220LE cameras equipped with Tamron 13VM2811ASIR lenses and visible light filters (750 nm high-pass filters) to capture the entire screen³. The cameras are positioned in portrait mode behind the screen, outside of the projection volume. The overlapped region between two cameras is about 30 cm large. The four cameras are connected to a PC (2.66 GHz Intel Xeon, 4 GB RAM, Windows 7) running a custom software that processes the four 752×480 images at 60 Hz and sends contact information to the machine in charge of the display over UDP.

Software additions

Starting from raw images like those of Figure 1, our software detects projected shadows and analyzes them to generate touch events. Like in a stadium where players have several shadows in different directions, the plurality of IR sources allows to discriminate contact points (where shadows converge) from the penumbra created by objects and people further away from the screen.

Our software uses the IDS SDK to adjust the settings (e.g. the exposure time) and retrieve images from the four cameras. The images are then processed using OpenCV, the cameras having been calibrated using a classical black-white chessboard. Each image is first corrected for barrel distortion. A background image taken while the room was empty is subtracted from it, which removes everything but the shadows (Figure 2, middle image). A Gaussian adaptive thresholding function with a 13×13 neighborhood is used to preserve only the darkest areas (Figure 2, right). Blobs are then extracted from these areas, filtered using size (minimum and maximum) and aspect-ratio thresholds, and geometrically corrected for on-screen projection. The blobs found in the four images are finally merged based on a distance threshold, labeled, and serialized as a TUIO bundle sent over the network.

The geometric correction for projection compensates the strong distortion caused by the curved shape of the screen and the positioning of the cameras near the floor. The curved screen is approximated by 10×6 flat rectangles, a classical homography being applied on each one to define the projection from the camera image to the screen.

Our system can detect 50+ simultaneous contacts with a precision between 3 and 5 mm (at the bottom and top of the screen) and an average overall latency of 140 ms (including 30 ms for image processing). With the configured thresholds, fingers are detected from 1 cm away from the

³The total cost for each camera is about 340 €

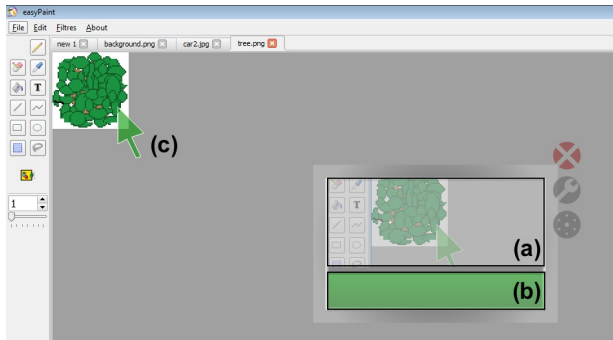


Figure 3. WallPad in a drawing application. The widget comprises a rectangular touch sensitive area (a) with a button below (b). Finger drags in the touch sensitive control the relative position of the pointer (c). The region around the pointer is displayed in the touch sensitive area. Each WallPad button and pointer have a unique color to help disambiguating them.

screen, which is hardly perceivable when touching an object but can be annoying when tapping as it requires larger finger movements. The calibrations of the cameras and projection are good enough so that users do not notice any offset between their finger(s) and the detected contact(s). The overall result is actually surprisingly good considering the small additions we made to the room equipment. Yet multi-touch sensing is not enough. Specific interaction techniques are also required to support usable multi-touch interactions with legacy (often 2D) and custom applications.

WALLPAD

WallPad⁴ is a new widget we designed to ease multi-touch interaction on wall displays. It supports the creation of proxy views in an easy and elegant way to bring distant interface regions within arm's reach. It is intended to help improving distant objects legibility through these views, ease switching between direct multi-touch interaction within them and indirect cursor-based interaction through them, and support fast and precise remote cursor manipulation through non-linear transfer functions. In the following, we describe the WallPad widget from its use as a regular touchpad to more sophisticated interactions, detailing its creation, moving, resizing and dismissal.

Regular Touchpad Use

To ease learning for novice users, WallPad was designed to operate by default as a conventional touchpad. Its visual representation is similar to one, comprising a rectangular touch sensitive area with a button below (Figure 3). As WallPad appears on top of existing applications, its representation is semi-transparent to reduce occlusion and visual clutter. Any finger drag in the rectangular area allows to relatively position a distant arrow-shaped pointer through a non-linear transfer function designed to easily cover large distances while preserving precise positioning (pointer movements are scaled down when movements are slow and scaled up when they are fast). We used a transfer

⁴the source code and video are available at <http://ns.inria.fr/mint/WallPad>

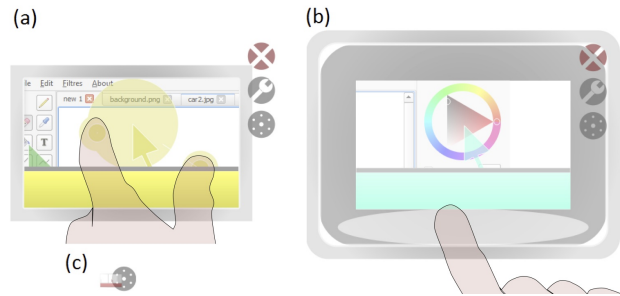


Figure 4. (a) WallPad screen capture in the de-localized direct interaction mode: WallPad is opaque to the background. (b) WallPad extended border after pressing the wrench button: the widget can be moved and resized. (c) Minimized view after pressing the minimized button.

function similar to the one used by OS X for real touchpads [4]. To reduce clutching after a drag has been initiated, finger tracking is not limited to the WallPad widget area during this operation. Taps on the button produce click events. As distant objects can be hard to perceive, the rectangular area displays a 1:1 view of the region underneath the pointer (clipped to the widget area, see Figure 3a) which allows to precisely position it, to select small objects or read small texts for example.

De-localized Direct Interaction

WallPad also allows to directly interact with the objects presented in the local rectangular view of the distant area (*de-localized direct interaction*). Single and double taps on the objects displayed in the rectangular region are passed to their remote location. Thus, users can directly select menu items or icons shown in the proxy view instead of having to first bring the pointer on the objects of interest and second pressing the button to select them (Figure 3a). Items are selected on touch release events to prevent selecting objects when the rectangular area is used as a regular touchpad.

Multi-touch object manipulation is enabled after holding a finger steady in the widget for 0.5s or touching it with several fingers. The view is turned from semi-transparent to opaque when multi-touch interaction is engaged to provide a visual feedback of the quasi-mode change and suggest objects can be further manipulated (Figure 4a). Contact areas are displayed as little colored disks around the pointer at the remote location to provide awareness to other users. WallPad remains in this mode for 0.5 s after the last contact is lost to support clutching while dragging objects.

WallPad LIFE CYCLE

The soft-aspect [11] of WallPad allows to create as many devices as needed, with different sizes and positions.

Creation gesture

WallPads can be dynamically created, moved, resized and dismissed. Creating one requires the use of an unambiguous gesture to prevent unexpected apparition. After considering different alternatives, we came up with the idea of bringing out a pointer from a finger. The corresponding gesture consists in first touching the display with a finger and, while keeping this first finger steady on the surface,

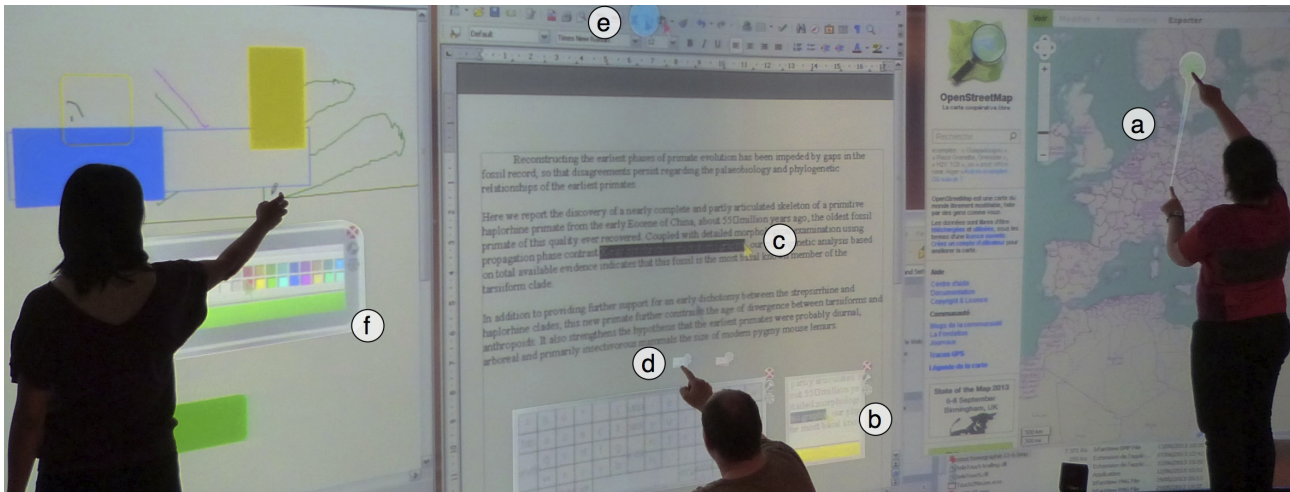


Figure 5. A user creating a WallPad (a). Another using two WallPads to select (b) some text (c) and click (d) on Copy/Paste buttons (e) beyond reach. A third user accessing a color palette located at the bottom of the screen through a WallPad (f).

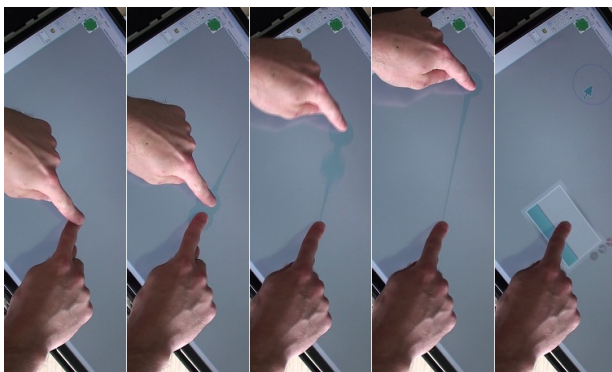


Figure 6. Creating a WallPad (shown here on a 3M multi-touch screen for clarity). From left to right: the user first slides a finger along a finger that remains steady on the surface and continues sliding on the display up to a distance threshold after which the WallPad is created. Feedback and feed-forward help knowing the gesture has been correctly recognized and predict when the WallPad will be created and where the pointer will appear.

using a finger from the other hand to slide on the first one up to the display and continue sliding for a given distance threshold (Figure 6). Once the sliding finger touches the display, specific feedback (a line and a disk) is used to indicate the gesture has been successfully recognized (Figure 5a). It is then used as feed-forward to show where the pointer will be created (Figure 6). The creation can be canceled by either removing the first finger or by not crossing the distance threshold with the second finger.

During our tests, we never faced ambiguous detection with other gestures such as pinch or rotate. This type of gesture can be easily extended to create other widgets. A virtual keyboard could be created by sliding two fingers instead of one, for example. The WallPad creation gesture could have interfered with the two finger scale gesture, but this one is usually performed by moving the fingers in opposite directions while our gesture expects the first finger to remain steady. The chosen gesture specifies both the desired widget location and the initial location of the cursor. It does not allow to create cursors in out-of-reach regions, but the relative non-linear mapping allows

to quickly move newly created cursors to these regions. A unique color is associated to each WallPad (button and pointer) to help disambiguate them.

Dynamic and reconfigurable aspects

A WallPad can be moved or resized after pressing the wrench icon to highlight a border around it (Figure 4b). Touching the border with one finger allows to move the WallPad while using two fingers allows to resize it. Resizing is especially useful when one wants the WallPad to fit the shape of a remote toolbar, for example (Figure 5f). It is also useful for adjusting the size of the working area when using the de-localized direct interaction (Figure 5b). The ability to create multiple local proxies for distant rectangular regions is somewhat similar to what *WinCuts* [17] and the *User Interface Façades* [16] support on desktop systems.

WallPads can be minimized by pressing a specific icon located below the wrench one (Figure 4c). Minimized WallPad appear as a small square buttons with a 20 mm side (Figure 5d). Once minimized, the corresponding pointer can no longer be moved (a drag gesture on a minimized widget moves it, not the corresponding pointer). However single and double tap events are still forwarded to it. Minimized WallPad are useful to reduce visual clutter and occasionally press a button which is far away such as undo/redo or copy/paste buttons (Figure 5d,e). Finally, a WallPad can be dismissed by hitting the cross button (Figure 4c).

Remote WallPad pointers are treated as ordinary graphical objects and can be directly manipulated. This is particularly useful when a pointer gets in the way of a remote user, by occluding some information for example. That user could simply move the pointer aside. This feature can also ease collaboration, a remote user having the possibility to position another user's pointer where he wants her to do something.

RELATED WORK AND DISCUSSION

Multi-touch walls afford co-localized interaction with screen content (*direct interaction*), a convenient way to

interact with objects within arm's reach. Yet interacting with farther objects quickly becomes tiring and inefficient because of the required body movements. To alleviate this problem a common solution is to interact indirectly with these remote objects using a pointer controlled with a relative mapping (*indirect interaction*). For example *HybridPointing* [7] was designed to switch from direct to indirect pen interaction on large displays by catching and releasing a trailing widget. Alternatively, objects can be thrown to distant locations but it can make precise positioning difficult [9]. The *SurfaceMouse* [1] emulates a desktop mouse on multi-touch surfaces to control a single cursor with a relative mapping. The Surface mouse is activated when the wrist and five fingers are in contact with the surface. Tangible tools have also been used to switch between direct and indirect interaction but they are not very practical with vertical surfaces [19]. However even if these techniques allow to switch between direct and indirect interaction using different methods, they only support single point interaction. The *Rizzo* virtual mouse allows to control a virtual pointer using a relative mapping while a magnified view of the remote objects is displayed near fingertips [18]. This improves the problem of legibility with distant objects but *Rizzo* was not designed to be used together with co-located interaction on the objects of interest nor does it support the manipulation of remote objects with multi-touch gestures.

Instead of controlling a cursor to interact with remote objects, *Drag&Pop* [2] and the *Vacuum* widget [3] allow to temporarily bring distant targets within arm's reach to interact with them. However, these techniques require knowledge about the objects of interest which makes them unsuitable for interaction in empty spaces, when drawing for example. Instead of providing access to selected remote objects, *WIM* [15] allows to interact with any distant object of a 3D scene by presenting a miniature view of the whole scene. In practice, interaction with small objects is rather difficult, especially in the case of large scenes of course. Back in the 2D world, rather than showing a miniature view of the whole workspace, *Frisbee* [10] brings a delimited (round) part of it within arm's reach. Users can directly manipulate remote objects through this duplicated view, without the need for a remote pointer. However the absence of a pointer at the remote location can actually be a problem in multi-user situations, if someone else is working in the same area. *Tablecloth* [13], which temporarily moves desktop portions, is also badly suited for multi-user situations as it would alter everyone's view of the display. Personal devices are sometimes used to support multi-user interaction with remote objects [5, 14, 12]. Yet interaction with a large display on a small separate input surface poses a lot of other questions [8].

Like the closest technique in the literature, *Frisbee*, *WallPad* allows to create, reposition and resize a local view on a remote space and to refine one's focus in that space. The main difference between the two lies in the interaction with distant objects. With *Frisbee*, since interaction can only take place in the local view, one has to make sure that all the objects and areas of interest for the task are locally visible beforehand. *WallPad* supports this mode of operation, but also pointer-based interaction in the re-

mote space. This in turn supports situated, unprepared interactions, and not just planned ones. The display of a pointer for action at the remote location instead of a disk shape corresponding to the local view's focus also has a profound impact on multi-user interaction. It reduces visual clutter and allows users to know what others are doing, and not just where they are generally looking. The use of multiple *Frisbees* is mentioned in [10], but only as a design enhancement "to support large-scale workspace interactions". Other advantages of *WallPad* over *Frisbee* include support for direct multi-touch interaction with remote objects, pointer-based indirect interaction using a non-linear mapping to reduce clutching (a 1:1 mapping is used when moving a *Frisbee's* focus in the remote space) and a rectangular shape better suited to common regions of interest (e.g. widgets and windows).

Using *WallPad*, users can dynamically create cursors whenever deemed necessary, each of them being independently controlled. The widgets are persistent - until users dismiss them - and are freely movable. *WallPads* can be used to repeatedly access remote content (e.g. widgets), to drag objects over large distances, or to interact with out-of-reach ones. By switching between different *WallPads*, users can quickly change the locus of interaction. Beyond the speed and accuracy advantages of using a pointer controlled through a non-linear transfer function, *WallPad* makes it possible to instantaneously move to different exact locations to support recurrent interactions, in a way similar to *UIMarks* [6].

CONCLUSION

WallPad was motivated by the lack of interaction techniques to properly support direct and indirect interaction on multi-touch wall displays. Throughout its design, it has been refined several times through informal user testing to obtain a tool easy to use in varied situations. The current version requires little learning. Once users know how to create a *WallPad* and how to transition to the de-localized direct interaction mode, the other operations are pretty straight-forward. *WallPad* is intended to reduce user fatigue when interacting with remote content and doing so without sacrificing precision or legibility. We expect *WallPad* to be a simple and powerful tool allowing users who master it to interact at a distance with applications that were not necessarily designed for large multi-touch displays.

ACKNOWLEDGEMENTS

This work was supported by the ANR project ANR-09-CORD-013 *InSTInCT* - <http://anr-instinct.cap-sciences.net> - and the Conseil Régional Nord - Pas de Calais.

REFERENCES

1. Bartindale, T., Harrison, C., Olivier, P., and Hudson, S. E. *SurfaceMouse*: supplementing multi-touch interaction with a virtual mouse. In *Proceedings of TEI '11*, ACM (2011), 293–296.
2. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A.

- Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proceedings of INTERACT '03* (2003), 57–64.
3. Bezerianos, A., and Balakrishnan, R. The vacuum: facilitating the manipulation of distant objects. In *Proceedings of CHI '05*, ACM (2005), 361–370.
 4. Casiez, G., and Roussel, N. No more bricolage! methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of UIST '11*, ACM (2011), 603–614.
 5. Chapuis, O., Bezerianos, A., and Frantzeskakis, S. Smarties: An input system for wall display development. In *Proceedings of CHI '14*, ACM (2014), 2763–2772.
 6. Chapuis, O., and Roussel, N. UIMarks: quick graphical interaction with specific targets. In *Proceedings of UIST '10*, ACM (2010), 173–182.
 7. Forlines, C., Vogel, D., and Balakrishnan, R. Hybridpointing: fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of UIST '06*, ACM (2006), 211–220.
 8. Gilliot, J., Casiez, G., and Roussel, N. Impact of form factors and input conditions on absolute indirect-touch pointing tasks. In *Proceedings of CHI '14*, ACM (2014), 723–732.
 9. Hascoët, M. Throwing models for large displays. In *Human Computer Interaction, HCI'2003*, B. H. Group, Ed. (Bath, UK, 2003), 77–108.
 10. Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N., and Kurtenbach, G. A remote control interface for large displays. In *Proceedings of UIST '04*, ACM (2004), 127–136.
 11. Nakatani, L. H., and Rohrlisch, J. A. Soft machines: a philosophy of user-computer interface design. In *Proceedings of CHI '83*, ACM (1983), 19–23.
 12. Nancel, M., Chapuis, O., Pietriga, E., Yang, X.-D., Irani, P. P., and Beaudouin-Lafon, M. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of CHI '13*, ACM (2013), 831–840.
 13. Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D. The large-display user experience. *IEEE Computer Graphics and Applications* 25, 4 (July 2005), 44–51.
 14. Satyanarayan, A., Weibel, N., and Hollan, J. Using overlays to support collaborative interaction with display walls. In *Proceedings of IUI '12*, ACM (2012), 105–108.
 15. Stoakley, R., Conway, M. J., and Pausch, R. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of CHI '95*, ACM (1995), 265–272.
 16. Stuerzlinger, W., Chapuis, O., Phillips, D., and Roussel, N. User interface facades: towards fully adaptable user interfaces. In *Proceedings of UIST '06*, ACM (2006), 309–318.
 17. Tan, D. S., Meyers, B., and Czerwinski, M. Wincuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI '04 Extended Abstracts*, ACM (2004), 1525–1528.
 18. Vlaming, L., Collins, C., Hancock, M., Nacenta, M., Isenberg, T., and Carpendale, S. Integrating 2D mouse emulation with 3D manipulation for visualizations on a multi-touch table. In *Proceedings of ITS '10*, ACM (2010), 221–230.
 19. Vogel, D., and Casiez, G. Conté: multimodal input inspired by an artist's crayon. In *Proceedings of UIST '11*, ACM (2011), 357–366.

Interaction par le regard : évaluation du retour d'information progressif

Van Bao NGUYEN

LIG, Université Grenoble-Alpes
BP53, 38041 Grenoble cedex 9, France
e-mail : baonv90@gmail.com

Francis JAMBON

LIG, Université Grenoble-Alpes
BP53, 38041 Grenoble cedex 9, France
e-mail : francis.jambon@imag.fr

Gaëlle CALVARY

LIG, Université Grenoble-Alpes
BP53, 38041 Grenoble cedex 9, France
e-mail : gaelle.calvary@imag.fr

Résumé

L'utilisation de l'oculométrie pour l'interaction par le regard se caractérise, dans le cas d'une approche monomodale, par un fort couplage entre perception et action : il n'est pas possible, dans le cas général, de distinguer une prise d'information d'une action de l'utilisateur. C'est pourquoi nous proposons une nouvelle approche, basée sur un retour d'information progressif, permettant une réduction de ce couplage. Les premiers résultats expérimentaux sont encourageants et indiquent que l'interaction par le regard peut être une technique d'interaction crédible dans certains contextes d'usage. Ils mettent aussi en lumière l'intérêt du retour d'information progressif.

Mots clés choisis par les auteurs

Oculométrie, interaction par le regard, Midas Touch, retour d'information progressif.

Mot clés de la classification ACM

H.5.2. [Information Interfaces and Presentation]: User Interface---Input devices and strategies, Interaction styles; I.3.6. [Computer Graphics]: Methodology and Techniques---Interaction techniques.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670463>

Problématique

Notre cadre d'étude s'intéresse à l'utilisation de l'oculométrie pour l'interaction homme-machine dans les situations où l'utilisateur ne souhaite pas utiliser ses mains, que ce soit pour des raisons de sécurité (e.g. chirurgien souhaitant consulter des radios sans utiliser ses mains pour garantir la stérilité du champ opératoire), de praticité ou de confort (e.g. personne consultant une recette de cuisine pendant que ses mains sont occupées à réaliser la dite recette).

En nous adressant a priori à tous les utilisateurs, notre approche se distingue des travaux concernant l'assistance des personnes avec handicap moteur. Notamment, nous nous intéressons à des usages de l'interaction par le regard qui peuvent être ponctuels, en complément d'une utilisation plus classique comme la souris ou le tactile. Ainsi, notre approche se doit de ne pas imposer des modifications profondes du paradigme d'interaction initial. Dans ce contexte, l'approche proposée se base également sur une interaction ne nécessitant pas d'apprentissage a priori.

En interaction par le regard, l'approche la plus naïve consiste à utiliser la position du regard dans le repère de l'écran comme un substitut direct pour le pointeur et le bouton d'une souris. Or, le regard n'est pas naturellement utilisé par l'humain pour agir sur l'environnement et plus généralement transmettre de l'information. Il est plutôt sollicité pour percevoir de l'information (sauf exception en relation humain-humain), de plus, certains mouvements du regard peuvent être la conséquence d'une activité réflexe.

Dans cette approche naïve, tout changement de position du regard de l'utilisateur entraînera le pointeur

de la souris à se déplacer, et chaque déplacement pourra lancer une nouvelle commande. Autrement dit, cette approche associe au regard une capacité d'action sur l'interface, et toute prise d'information pourra se voir interprétée comme une action. C'est le verrou du "Midas Touch" [3] inspiré du Mythe de Midas qui transformait en or tout ce qu'il touchait, et ainsi ne pouvait plus ni manger ni boire.

Ainsi, l'utilisation naïve du regard comme dispositif d'entrée provoque un très fort couplage entre la prise d'information et l'action. L'objectif de nos travaux est de permettre l'utilisation du regard comme mode d'interaction unique, tout en contournant le verrou du "Midas Touch". Autrement dit, il s'agit de déterminer un paradigme pour assurer le découplage entre la perception et l'action lorsque l'on souhaite utiliser l'oculométrie comme un moyen unique d'interaction.

État de l'art

Les recherches en oculométrie remontent à la fin du XIXe siècle lorsqu'en 1879, Louis Javal a découvert des mouvements saccadés au niveau de l'œil dans une tâche de lecture. En 1908, Edmund Huey [1] a construit le premier oculomètre permettant de suivre le mouvement des yeux d'une personne en train de lire. Dans les années 50, Alfred L. Yabus [9] a défini ces mouvements en termes de fixations et de saccades, concepts toujours utilisés actuellement.

L'utilisation de l'oculométrie pour l'interaction homme-machine n'a vraiment été étudiée qu'au cours de ces trente dernières années, pendant lesquelles nous pouvons identifier cinq étapes remarquables :

En **1982**, Richard A. Bolt [2] a été le pionnier en proposant plusieurs approches pour utiliser les mouvements des yeux en interaction homme-machine. L'une des approches proposées nécessitait de faire appel à la reconnaissance vocale pour désambigüiser certaines des commandes issues du regard.

En **1991**, Jacob et al. [3] ont utilisé l'oculométrie pour implémenter un système interactif comportant les principales interactions (sélection, déplacement, scroll, menu, etc.). Ils ont également identifié le problème du couplage perception/action et c'est à ces auteurs que l'on doit la dénomination de « Midas Touch ». Jacob et al. ont proposé plusieurs techniques d'interaction combinant le regard et l'utilisation du clavier, ou le regard seul par la technique du « dwell » (déclenchement lié à la durée de fixation). Leurs travaux ont posé les bases de l'interaction par le regard. À la suite de Jacob et al., d'autres chercheurs ont travaillé sur l'utilisation du clavier et de la souris pour éviter le « Midas Touch », comme Vetegaal [4] ou Kumar et al. [5].

En **1999**, Zhai [6] a introduit la technique de « Magic Pointing », dédiée à améliorer la précision du pointage à la souris. En « Magic Pointing », le pointeur est automatiquement déplacé vers la zone dans laquelle l'utilisateur regarde. Il s'agit d'une approche un peu différente des précédentes, car ici le regard intervient non pas comme principale technique d'interaction, mais comme assistance pour améliorer la performance.

En **2007**, Drewes et Schmidt [7] ont mis en œuvre une interaction visuelle qui prend en compte les positions que l'utilisateur regarde continuellement, puis cherche à classer les « gaze gestures » à partir des

mouvements du regard depuis cette position vers l'une des 8 directions de sortie possibles.

En **2013**, Biswas et al. [8] ont proposé une nouvelle forme d'interaction destinée aux handicapés moteurs en employant une variante du « blink » (déclenchement lié au clignement des yeux) pour réaliser des tâches comme la sélection d'objet ou l'activation de commande.

En synthèse, l'interaction par le regard a fait l'objet de nombreux travaux depuis plus de trente ans, sans pour autant permettre à cette technique d'interaction de se diffuser dans le grand public. En effet, il existe un consensus parmi les auteurs pour présenter le « Midas Touch » comme principal verrou de la technique. Les auteurs ont répondu à cette problématique via des approches multimodales (voix, clic souris, touche clavier), ou monomodales (« blink », « dwell », « gaze gestures »). Cependant, aucune de ces approches ne s'est généralisée.

Contribution

Une action de l'utilisateur dans un environnement d'interaction graphique de type WIMP peut généralement se scinder en deux informations : une position donnée, par exemple, par les coordonnées du curseur de la souris, et un déclencheur, qui peut être l'appui sur le bouton de cette même souris.

Si l'on cherche à déterminer ces deux informations en interaction monomodale par le regard, la position sur l'écran est déduite assez simplement par la position du regard. Concernant le déclencheur, la littérature propose notamment le « dwell » ou le « blink ». Nous avons écarté a priori le déclencheur « gaze gesture » car il implique l'apprentissage d'une grammaire de

gestes, ce qui n'est pas adapté à notre contexte d'usage ponctuel.

Le « dwell » consiste à déclencher une commande si l'utilisateur maintient la position de son regard fixe pendant un temps donné, dans une zone donnée de l'écran. Le « blink » consiste à lier le déclenchement à un clignement des yeux lorsque le regard de l'utilisateur est à l'intérieur d'une zone donnée de l'écran. Ces deux techniques ont leurs limites, et des déclenchements non désirés peuvent se produire très facilement lorsque l'utilisateur regarde trop longtemps un bouton de commande, ou s'il cligne des yeux de manière réflexe en regardant une commande.

Notre hypothèse est que le fonctionnement du déclencheur n'est pas assez explicite pour l'utilisateur. En effet, les retours d'information proposés dans la littérature se contentent d'informer l'utilisateur au moment du déclenchement de l'action, comme cela se produit lorsque l'on clique sur un bouton avec une souris. Nous supposons qu'il est nécessaire de ne pas se limiter à la seule indication du déclenchement, mais d'apporter un retour d'information à grain fin sur l'ensemble des étapes du processus d'interaction.

C'est pourquoi, nous proposons d'apporter un retour d'information de type progressif qui concrétise visuellement les étapes et le temps requis/restant pour activer une commande ou sélectionner un élément dans une liste. Nous avons appliqué ce principe aux deux types de déclencheurs « dwell » et au « blink ».

Application au déclencheur « dwell » (figure 1) : Ce déclencheur est bien adapté aux actions qui ne nécessitent pas une lecture de longue durée du label de l'action, car ainsi, le temps de lecture requis peut être

inférieur au seuil de déclenchement. Ce déclencheur peut être utilisé, par exemple, pour activer un bouton. Dans notre implémentation, l'activation est déclenchée, si, une fois passé le seuil de sélection à T_{fix} (correspondant à la durée minimale d'une fixation), l'utilisateur maintient son regard sur l'objet jusqu'à T_{init} .

Un retour d'information progressif (sous forme de barre de progression située à l'intérieur du bouton) est utilisé entre T_{fix} et T_{init} de manière à permettre à l'utilisateur de visualiser le temps restant jusqu'à l'activation (sans quitter des yeux le bouton). L'utilisateur peut alors, en cours d'interaction, porter son regard en dehors de l'objet et ainsi ne pas déclencher l'action.

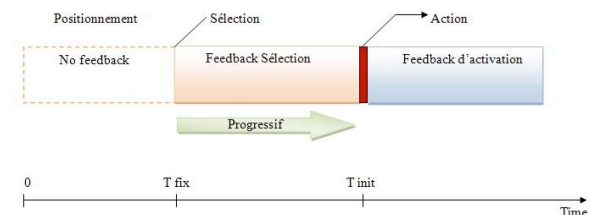


Figure 1. Déclencheur « dwell » avec retour d'information progressif.

Application au déclencheur « blink » (figure 2) : Ce déclencheur est bien adapté aux actions nécessitant une lecture de longue durée du label de l'action, car sa période d'activation peut être bornée. Il peut être utilisé, par exemple, pour sélectionner un élément dans une liste. Le principe de ce déclencheur est de ne permettre une activation par un clignement des yeux que pendant un intervalle de temps limité, afin d'éviter un déclenchement involontaire sur une activité réflexe. Dans notre implémentation, l'activation n'est possible qu'entre T_{init} et T_{exp} , et pendant cet intervalle de temps,

un retour d'information progressif (sous forme de décompte situé à l'intérieur du label) indique à l'utilisateur combien de temps il lui reste pour effectuer un clignement des yeux (sans quitter des yeux le label), et ainsi déclencher l'action (figure 2 : cas b) ou pas (figure 2 : cas a).

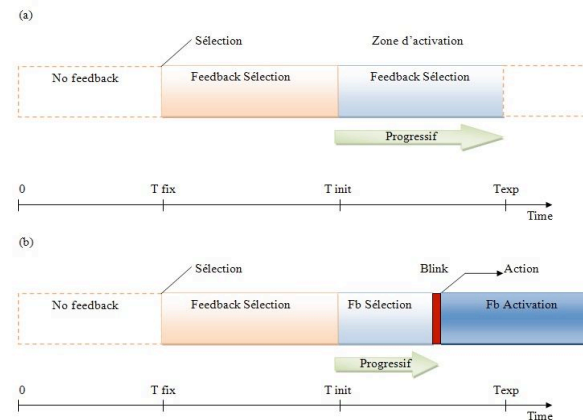


Figure 2. Déclencheur « blink » avec retour d'information progressif.

Nous avons mis en œuvre ces deux techniques d'interaction sur un démonstrateur : l'application EyeMail. Cette application est un logiciel simplifié de lecture de courriers électroniques, qui permet d'accomplir les principales tâches de ce type de logiciel : la sélection d'un courrier dans la boîte de réception, la composition d'un nouveau courrier, l'ajout d'une image jointe, son envoi, etc. Toutes ces commandes peuvent être réalisées en utilisant uniquement le regard. Nous avons ainsi employé les deux techniques d'interaction précédemment proposées :

- Activation par « dwell » : nous avons utilisé ce déclencheur pour activer les commandes (composer un e-mail, envoyer un e-mail ou attacher une image) ;
- Sélection par « blink » : nous avons utilisé ce déclencheur pour sélectionner un élément dans une liste (sélection des e-mails, sélection des images).

Évaluation

Nous avons réalisé des expérimentations utilisateur afin d'obtenir une première évaluation de l'apport effectif du retour d'information progressif. Nous avons testé les deux techniques d'interaction « dwell » et « blink », chacune avec et sans retour d'information progressif. Dans les deux cas, nous avons conservé le retour d'information présent lors du déclenchement de la commande. Pour ces expérimentations, nous avons utilisé trois logiciels : l'application EyeMail pour l'habituation, une application « Groupes de boutons » pour évaluer le « dwell » (le sujet doit activer le bouton indiqué dans la consigne parmi les 8 boutons disponibles), et une application « Liste d'éléments » pour évaluer le « blink » (le sujet doit sélectionner l'élément indiqué dans la consigne parmi une liste de 8 éléments disponibles).

Le protocole expérimental débutait par une calibration de l'oculomètre, puis le sujet pouvait s'habituer avec l'application EyeMail pendant le temps qu'il souhaitait. Ensuite, il effectuait successivement 4 tests différents : avec chacune des deux techniques, et avec/sans retour d'information progressif. L'ordre des techniques comme celui de la présence du retour d'information progressif étaient alternés. Les résultats ont été analysés avec les données suivantes : nombre d'activations réussies, nombre d'activations en dehors de la consigne, et avis subjectif des utilisateurs. En tout, 13 sujets,

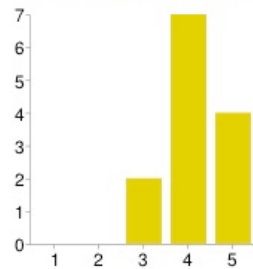


Figure 3. Avis des utilisateurs concernant la facilité d'utilisation du « dwell ».

principalement des étudiants et des chercheurs en informatique, ont participé aux expérimentations.

Tout d'abord, concernant la technique du « dwell », 12 sujets sur 13 ont préféré la présence d'un retour d'information progressif, ils ont tous donné un avis neutre ou favorable à la facilité d'activation des commandes par cette technique (figure 3). La note moyenne est de 4,15 sur un barème de 1 à 5 (5 étant le plus favorable). Concernant le déclenchement par « blink », les sujets ont également majoritairement préféré le feedback progressif : 10 sujets sur 13. La note moyenne donnée par les utilisateurs a été de 3,7 (figure 4). Les résultats sont globalement moins positifs que pour la technique du « dwell ».

Enfin, en prenant en compte le nombre d'activations en dehors des cibles dans les deux modes (avec et sans retour d'information progressif), les interactions avec retour d'information progressif se montrent généralement plus efficaces que celles sans retour d'information : les activations en dehors de la cible sont au total de 4 avec retour d'information progressif et de 8 lorsque l'on ne l'utilise pas.

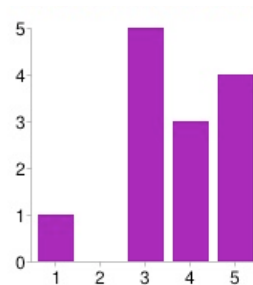


Figure 4. Avis des utilisateurs concernant la facilité d'utilisation du « blink ».

Conclusion & Perspectives

Bien que le nombre de sujets ne soit pas significatif et qu'ils ne soient pas représentatifs de la population générale, ces premiers résultats sont encourageants et indiquent que l'interaction par le regard est une technique crédible dans certains contextes d'usage, et que la présence d'un retour d'information progressif semble être préférée par les utilisateurs.

Ces premiers résultats devront cependant être confirmés par la mise en œuvre de nouvelles expérimentations comportant un panel plus important

d'utilisateurs et en variant les conditions de l'expérimentation. Notamment, il serait nécessaire de développer des applications spécifiques à un domaine particulier (santé, domotique, etc.) et de les évaluer en situation écologique.

Références

- [1] Huey, E. B. *The Psychology and Pedagogy of Reading* (Reprint). MIT Press (1968 originally published 1908).
- [2] Bolt, R. A. Eyes at the interface. In proc. *CHI'82*. ACM (1982), 360-362.
- [3] Jacob, R. J. K. What you look at is what you get: eye movement-based interaction techniques. In proc. *CHI'90*, ACM (1990), 11-18.
- [4] Fono, D., and Vertegaal, R. EyeWindows: evaluation of eye-controlled zooming windows for focus selection. In proc. *CHI'05*, ACM (2005), 151-160.
- [5] Kumar, M., Paepcke, A., and Winograd, T. EyePoint: practical pointing and selection using gaze and keyboard. In proc. *CHI'07*, ACM (2007), 421-430.
- [6] Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In proc. *CHI'99*, ACM (1999), 246-253.
- [7] Drewes, H., and Schmidt, A. The MAGIC touch: Combining MAGIC-pointing with a touch-sensitive mouse. In proc. *INTERACT'09*, Springer-Verlag (2009), 415-428.
- [8] Biswas, P. and Langdon, P. A new input system for disabled users involving eye gaze tracker and scanning interface. *Journal of Assistive Technologies*, vol. 5, iss. 2 (June 2011), 58-66.
- [9] Yarbus, A.L. *Eye Movements and Vision*. New York: Plenum press (1967).

Un dispositif d'interaction pour favoriser la discussion et l'engagement dans les musées

Patricia Plenacoste
Université Lille 1 - LIFL
F-59655 Villeneuve d'Ascq,
France
patricia.plenacoste@univ-
lille1.fr

Yvan Peter
Université Lille 1 - LIFL
F-59655 Villeneuve d'Ascq,
France
yvan.peter@univ-lille1.fr

Résumé

Les musées proposent de plus en plus de dispositifs numériques destinés à enrichir les visites. Toutefois, ceux-ci peuvent accaparer outre mesure l'attention des visiteurs et les isoler socialement. Nous proposons un dispositif destiné à des groupes de visiteurs. Il offre une interaction peu intrusive avec l'objectif de favoriser l'implication des visiteurs ainsi que les discussions entre ceux-ci. Ce dispositif a été déployé et expérimenté dans un musée des Beaux Arts. Nous décrivons la conception du prototype ainsi que l'expérimentation mise en place et les résultats. Celle-ci a permis de valider l'acceptabilité et l'utilisabilité du dispositif et montre un effet positif en terme d'attention portée aux œuvres et dans une moindre mesure de génération de discussions entre les visiteurs.

Mots-clés

visite de musée; collaboration; engagement; dispositif d'interaction ; interaction sociale

ACM Classification Keywords

H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces : Input devices and strategies, Prototyping, Evaluation/methodology.

General Terms

Musée, Interaction, Collaboration

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ème} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670464>

Introduction

Les conservateurs de musées cherchent à fournir une expérience riche et engageante à leurs visiteurs. Dans cette optique, les technologies d'interaction enrichissent l'information concernant les œuvres et l'expérience de visite. Parmi les technologies déployées dans les musées on peut citer les guides interactifs [5][8], les écrans et tables tactiles [7] ainsi que la réalité augmentée [1][2]. Elles offrent l'opportunité de répondre aux attentes des visiteurs en terme de facilité d'accès et d'expérience ludique [11] et permettent une plus grande implication du visiteur vis à vis des œuvres. Toutefois, elles induisent un *effet d'accaparement*. L'interaction avec le dispositif capture l'essentiel de l'attention du visiteur au détriment de l'expérience de visite et/ou de l'interaction avec les autres visiteurs [10]. Ce phénomène peut être observé avec des technologies différentes, du guide interactif [8] à la réalité augmentée [2].

La collaboration, le partage et l'expérience sociale sont une autre dimension de l'engagement des visiteurs avec les expositions. Vom Lehn affirme que, «l'interaction sociale et la collaboration sont de plus en plus importantes pour la conception et la mise en place des œuvres et des expositions.» [12]. En effet, la discussion entre les visiteurs permet l'échange de points de vue et amène à élaborer les raisons de son analyse ou de son ressenti face à une œuvre ce qui favorise l'apprentissage et une expérience de visite plus approfondie [6].

Les mécanismes de collaboration offerts au sein de l'exposition doivent être facilement compréhensibles pour permettre la réussite des interactions sociales [6]. Elliston & Fitzgerald proposent un guide de musée présenté sous la forme d'un dialogue informel et spontané entre deux conservateurs. Ils fournissent ainsi un contexte partagé

pour les interactions et réactions de paires de visiteurs [4]. Cosley et al. intègrent à leur application la possibilité d'ajouter des annotations sur les œuvres. Ces dernières donnaient un sentiment de connexion avec les visiteurs précédents et offraient l'opportunité de réfléchir aux œuvres de façon nouvelle [3]. Laaksolahti et al. ont proposé un dispositif ad hoc permettant de laisser et de retranscrire des traces tactiles localisées sous forme de vibrations variant en durée et en intensité [9]. Ce dispositif fournit donc également un mécanisme de partage entre des visiteurs successifs.

Notre proposition est focalisée sur l'utilisation des technologies d'interaction pour favoriser l'implication des visiteurs vis à vis des œuvres, sans effet d'accaparement de l'attention, et sur la dimension sociale de la visite de musée. Pour cette étude, nous avons conçu une borne question permettant aux visiteurs de choisir entre deux réponses possibles. L'objectif est d'inciter les visiteurs à se regrouper et à entrer en discussion à propos des œuvres. Dans la section suivante, nous présenterons la conception du dispositif. Ensuite nous décrirons la mise en œuvre. Après cela, nous introduirons le cadre expérimental ainsi que la méthodologie suivie avant de présenter les résultats de cette première phase d'expérimentation.

Conception du dispositif

Les éléments présentés dans l'introduction nous ont conduit à définir les conditions suivantes pour la conception de notre dispositif :

- C1 - Social** L'interaction sociale entre les visiteurs est le but premier du dispositif proposé.
- C2 - Discret** Le dispositif ne doit pas accaparer outre mesure l'attention des visiteurs.
- C3 - Simple** L'usage et les mécanismes d'interaction proposés doivent être facile à comprendre.

Le dispositif développé est une borne question proposant de choisir entre deux réponses possibles. La réponse choisie est sélectionnée en apposant un tag RFID dessus. Ce tag étant partagé au sein du groupe de visiteurs afin de les amener à négocier la réponse choisie. Suivant la condition C1, le dispositif proposé vise uniquement à déclencher une interaction sociale entre les visiteurs (discussion, échange de point de vue) autour d'une ou plusieurs œuvres. Ce choix est également cohérent avec la condition C2 dans la mesure où l'interaction entre les visiteurs est directe. Le mécanisme d'interaction est simple à comprendre (C3) et l'interaction est ponctuelle et rapide limitant ainsi l'attention portée au dispositif (C2).

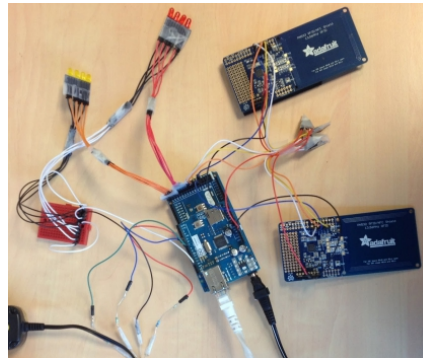


Figure 1 – Décomposition du prototype de borne.

Prototype

Les bornes questions sont basées sur un micro contrôleur Arduino Mega permettant de prendre en charge deux lecteurs RFID correspondant aux réponses possibles (figures 1 et 2). Les réponses précédentes sont indiquées à l'utilisateur sous forme de barres de LEDs correspondant au pourcentage de choix pour chaque réponse. Ces choix

de conception visent les objectifs suivants :

- l'utilisation d'un tag RFID, de préférence à une interaction tactile ou à un simple bouton, vise à faire partager à un groupe de visiteur le moyen de choisir la réponse, ceci afin de favoriser la discussion sur ce choix.
- l'indication des choix antérieurs a pour objectif de créer un lien avec les visiteurs précédents comme cela a pu être proposé dans les travaux cités précédemment [3][9].

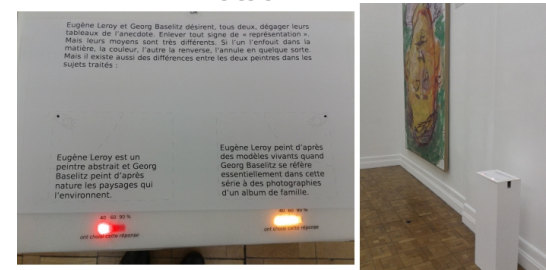


Figure 2 – Prototype de borne dans le musée.

Expérimentation

Les effets attendus du dispositif proposé sont de contribuer à l'engagement des visiteurs vis à vis de l'exposition et à la dimension sociale de la visite en leur permettant de confronter leurs avis et de discuter. En particulier, nous souhaitons vérifier les hypothèses suivantes :

- H1 Le dispositif proposé a un impact sur l'engagement des visiteurs vis à vis des œuvres (changement de perspective, revoir les œuvres...).
- H2 Le dispositif proposé a un impact sur la dimension sociale de la visite en générant des discussions au sein des groupes de visiteurs.

H3 Le dispositif est facile à utiliser et n'accapare pas l'attention des visiteurs.

Cadre Expérimental

Le dispositif est déployé et expérimenté dans un musée des beaux arts régional : MUba Eugène Leroy | Tourcoing, France. Le MUba expose des œuvres modernes et contemporaines avec une approche laissant la place aux sentiments et à l'appréciation personnelle plutôt qu'une approche plus encyclopédique. Pour cette raison, il y a peu d'informations fournies à propos des œuvres. Toutefois un dépliant est disponible à l'entrée du musée permettant aux visiteurs de trouver les informations correspondant à chacune des œuvres exposées.

Méthodologie

Une première phase d'expérimentation a eu lieu deux dimanches afin de maximiser le nombre de visiteurs. Trois bornes ont été réparties dans des salles différentes avec des questions en lien avec les œuvres proches. Un tag RFID était proposé aux groupes de visiteurs. Une brève explication du fonctionnement des bornes était donnée aux visiteurs. Aucune consigne particulière ne leur a été donnée quand à leur utilisation.

Nous avons capturé des traces correspondant aux interactions avec les bornes et nous avons réalisé des observations sur un certain nombre de groupes. Par ailleurs, un questionnaire de Likert a été soumis aux visiteurs après les visites. Celui-ci comprenait 15 questions couvrant les aspects suivants : acceptabilité (4 questions), utilisabilité (4 questions), impact sur la visite (3 questions), lien avec les visiteurs précédents (4 questions). Pour chaque question, quatre réponses étaient possibles (pas du tout d'accord - pas d'accord - d'accord - tout à fait d'accord). Les questions relatives au même thème étaient mélangées et des formulations négatives et

positives ont été utilisées. Par ailleurs, une partie réponse libre en fin de questionnaire était utilisée pour les remarques et suggestions.

Résultats

Les résultats présentés portent sur la deuxième session d'expérimentation. En effet, suite à la première journée d'expérimentation, nous avons dû faire évoluer la présentation des bornes pour régler certains problèmes d'utilisabilité (e.g., identification claire de la zone de détection RFID) ainsi que le questionnaire.

Nous avons obtenu 22 questionnaires pour une moyenne d'âge de 41 ans et 77,27% de femmes. Sur cette journée d'expérimentation, la plupart des groupes étaient composés de 2 ou 3 personnes.

Acceptabilité et utilisabilité

Plus de 86% des visiteurs seraient d'accord ou tout à fait d'accord pour réutiliser la borne lors d'une autre visite. Plus de 86% souhaiteraient la retrouver dans d'autres musées. Nous pouvons également noter un effet d'engagement du dispositif d'interaction dans la mesure où seulement 31% des visiteurs pensent que les questions seules sans dispositif d'interaction suffiraient. L'utilisation est simple pour plus de 90% des utilisateurs, bien que la rétroaction suite à la sélection d'une réponse doive être améliorée car actuellement trop discrète.

Impact sur la visite

Nous avons cherché à savoir si la présence des bornes avait changé la visite (i) dans l'approche des œuvres ; (ii) dans la dimension sociale de discussion entre visiteurs. Sur le premier point on observe un effet non négligeable dans la mesure où plus de 59% des visiteurs indiquent avoir été amenés à revoir les œuvres pour répondre aux questions. Ils sont également plus de 86% à penser que cela donne

un éclairage nouveau à leur perception des œuvres. Ces résultats confirment les observations réalisées. En effet, on note que les groupes font le tour des œuvres de la salle, lisent la question sur la borne puis retournent voir certaines œuvres avant de sélectionner une réponse. On retrouve cela également dans les commentaires libres :

- «... je n'avais pas fait attention à certains détails» (Femme, 31 ans);
- «...apporte une vraie relecture des œuvres proposées» (Homme, 24 ans);
- «...nous sommes amenés à réfléchir en approfondi sur les différentes œuvres proposées» (Femme, 45 ans).

L'effet est plus mitigé sur la dimension discussion puisque seulement 22% des visiteurs indiquent avoir discuté pour répondre. Toutefois, la formulation de la question devra être changée ("Les questions m'ont amené à discuter avec d'autres visiteurs pour répondre") car elle semble avoir été interprétée comme portant sur des discussions hors du groupe. Cela est notable pour une personne ayant répondu qu'elle n'a pas discuté alors qu'elle indique dans les commentaires libre du questionnaire qu'elle a apprécié «[l']échange dans le couple» (Femme, 29 ans). Par ailleurs, nous avons observé des situations d'échange dans différents groupes : parent - enfant, groupes d'amis.

Lien avec les visiteurs précédents

Plus de 90% des visiteurs ont compris que les barres de LEDs indiquent les choix des visiteurs précédents. Mais, seulement 36% des visiteurs ont indiqué avoir été influencés par les réponses précédentes.

Par ailleurs, seulement 50% des visiteurs pensent que leur réponse laisse une trace pour les visiteurs suivants. Ce résultat, en contradiction avec la première réponse peut probablement s'expliquer par le fait que le pourcentage de

réponse présente une vision agrégée et ne laisse pas au visiteur le sentiment que sa réponse sera visible par les suivants.

Conclusion

Notre hypothèse principale est que des dispositifs d'interaction simples peuvent solliciter le visiteur de manière à approfondir sa visite et à en renforcer la dimension sociale. Nous voulons éviter l'effet d'accaparement qu'on peut observer avec des dispositifs plus riches sur le plan informationnel mais plus complexes. Pour cela, nous avons conçu une borne question qui a été déployée dans un musée des Beaux Arts.

L'expérimentation montre que le système est bien accepté et qu'il est facile à utiliser évitant cet effet d'accaparement (hypothèse H3). La dimension interaction du dispositif est vue comme un apport par 69% des visiteurs. On constate également un impact sur la visite dans la mesure où il amène les visiteurs à revoir certaines œuvres et à les reconsidérer afin de répondre aux questions (hypothèse H1). Pour une partie des groupes, il génère effectivement des échanges avant de répondre, ce qui renforce ainsi la dimension sociale de la visite (hypothèse H2).

Pour l'instant, les questions fournies par le musée proposaient une seule bonne réponse. Il serait intéressant de disposer de questions pour lesquelles les deux réponses sont possibles afin de voir l'impact sur la discussion et l'utilisation du tag RFID (e.g., plusieurs votes différents au sein d'un groupe).

Nous aurions souhaité renforcer également la dimension sociale avec les visiteurs précédents. L'indication des pourcentages des réponses précédentes n'a visiblement pas eu d'effet significatif en ce sens. Il sera nécessaire de réfléchir à d'autres indicateurs de l'activité passée dans le

musée et de leur prise en compte dans l'interaction pour renforcer cette dimension.

Remerciements

Ce travail a été financé par Pictanovo (<http://www.pictanovo.com>) et la région Nord-Pas de Calais. Nous remercions le musée MUba Eugène Leroy | Tourcoing, particulièrement Sueva Lenôtre, Laure Perret et Evelyne-Dorothee Allemand pour leur implication. Nous remercions également Romain Giovanetti pour le développement et la mise au point des prototypes.

Références

- [1] Barry, A., Thomas, G., Debenham, P., and Trout, J. Augmented reality in a public space : The natural history museum, london. *Computer* 45, 7 (2012), 42–47.
- [2] Chang, K.-E., Chang, C.-T., Hou, H.-T., Sung, Y.-T., Chao, H.-L., and Lee, C.-M. Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum. *Computers & Education* 71 (2014), 185–197.
- [3] Cosley, D., Baxter, J., Lee, S., Alson, B., Nomura, S., Adams, P., Sarabu, C., and Gay, G. A Tag in the Hand : Supporting Semantic, Social, and Spatial Navigation in Museums. Dans *Actes de CHI* (2009), 1953–1962.
- [4] Elliston, B., and FitzGerald, E. Encouraging museum visitor engagement using spontaneous talk-in-interaction audio guides. Dans *Actes de CSEDU* (2012).
- [5] Fleck, M., Frid, M., Kindberg, T., Rajani, R., O'Brien-Strain, E., and Spasojevic, M. From Informing to Remembering : Deploying a Ubiquitous System in an Interactive Science Museum. *Pervasive Computing* 1, 2 (2002), 13–21.
- [6] Heath, C. Interaction and interactives : collaboration and participation with computer-based exhibits. *Public Understanding of Science* 14, 1 (Jan. 2005), 91–101.
- [7] Hornecker, E. "I don't understand it either, but it is cool" - visitor interactions with a multi-touch table in a museum. Dans *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, IEEE (Oct. 2008), 113–120.
- [8] Hsi, S. A study of user experiences mediated by nomadic web content in a museum. *Journal of Computer Assisted Learning* 19, 3 (Sep. 2003), 308–319.
- [9] Laaksoaho, J., Tholander, J., Lundén, M., Solsona Belenguer, J., Karlsson, A., and Jaensson, T. The lega : a device for leaving and finding tactile traces. Dans *Actes de TEI*, ACM (2011), pp. 193–196.
- [10] Lyons, L. Designing opportunistic user interfaces to support a collaborative museum exhibit. Dans *Actes de CSCL* (2009), 375–384.
- [11] Sheng, C.-W., and Chen, M.-C. A study of experience expectations of museum visitors. *Tourism Management* 33, 1 (2012), 53–60.
- [12] vom Lehn, D., Heath, C., and Hindmarsh, J. Video-based Field Studies in Museums and Galleries. *Visitor Studies Today* V, III (2002), 15–23.

Quick-Glance and In-Depth exploration of a tabletop map for visually impaired people

Sandra Bardot

IRIT, CNRS & University of Toulouse
sandra.bardot@irit.fr

Anke Brock

Inria Bordeaux
anke.brock@inria.fr

Marcos Serrano

IRIT, CNRS & University of Toulouse
marcos.serrano@irit.fr

Christophe Jouffrais

IRIT, CNRS & University of Toulouse
christophe.jouffrais@irit.fr



Abstract

Interactive tactile maps provide visually impaired people with accessible geographic information. However, when these maps are presented on large tabletops, tactile exploration without sight is long and tedious due to the size of the surface. In this paper we present a novel approach to speed up the process of exploring tabletop maps in the absence of vision. Our approach mimics the visual processing of a map and consists in two steps. First, the Quick-Glance step allows creating a global mental representation of the map by using mid-air gestures. Second, the In-Depth step allows users to reach Points of Interest with appropriate hand guidance onto the map. In this paper we present the design and development of a prototype combining a smartwatch and a tactile surface for Quick-Glance and In-Depth interactive exploration of a map.

Author Keywords

Map exploration, visual impairment, accessibility, tabletop, smartwatch, spatial cognition

ACM Classification Keywords

K.4.2. [Computers and Society]: Social Issues - Assistive Technologies for Persons with Disabilities; H.5.2. Information interfaces and presentation: User Interfaces- Input Devices and strategies.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670465>

Introduction

For visually impaired people, it is a challenge to understand the surrounding and to navigate autonomously, which significantly reduces mobility and independence. Preliminary exploration of a tactile map of the environment [12] and preparation of the journey may both improve autonomy. However, tactile maps present important limitations. First, in order to improve tactile perception and reduce cognitive load, the amount of information on the map is limited. It is also limited by the fact that Braille legend takes a lot of space. Second, once the maps are printed, the content cannot be adapted dynamically. Tactile maps are thus quickly out of date [15]. Free digital online maps (for example Open Street Map) are nowadays largely available, but are most often not accessible to visually impaired people. Dedicated accessible interactive maps have been developed based on different interactive devices [2]. It has been shown that interactive maps enable visually impaired people to efficiently acquire knowledge about unknown environments [3]. However there is still little research work aiming to provide visually impaired users with advanced non-visual interaction techniques for map exploration.

The goal of our project is to speed up the exploration process of a large tactile interactive map with non-visual interaction techniques. Although the use of large tabletops is more and more common, it has so far been rarely explored for presenting accessible maps to visually impaired people [5]. The large size of the screen induces increased exploration time, and thus impacts interaction [4], behavioral and cognitive processes.

In this work we propose a two steps non-visual interaction that mimics the visual processing of a map. The first step (Quick-Glance) allows quickly acquiring a global but sparse mental representation of the map by using mid-air gestures. In a second step (In-Depth), users can select a point of interest (POI) to which the hand is guided. The interaction is provided via the combination of a smartwatch and a multi-touch tabletop. In this paper we present the design and implementation of the prototype, and propose future directions for evaluations.

Related work

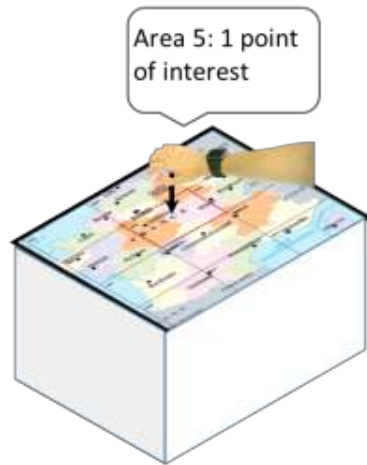
Interactive Maps for visually impaired people

In the past years interactive maps dedicated to visually impaired people have been developed based on the use of different interactive technologies such as touch screens, haptic feedback, speech recognition, mouse, keyboard and/or tangible interaction [2]. Many prototypes are based on the use of different types of touch-sensitive devices combined with audio feedback. We focus here on those interactive maps providing touch-based interaction. Touch-screens of the size of a regular computer screen have been used in several prototypes [3,7,14]. Fewer projects were based on large tabletops [5]. Recently the use of mobile devices, such as smartphones and tablets, has emerged [9,15]. In some accessible map prototypes, the touch-sensitive surface was combined with raised-line map overlays [7] or with vibro-tactile stimulation [9,15] in order to add tactile information which is not provided by the touch screen per se.

Interaction Techniques for on-Map Hand Guidance

The above-mentioned prototypes generally provide the possibility to obtain information about points of

Step 1. Quick-Glance exploration



Step 2. In-Depth guidance



Figure 1. Illustration of the 2-step process: Quick-Glance exploration followed by In-Depth guidance.

interests (POI) and streets by executing finger taps. Kane et al. [5] proposed three techniques for assisting visually impaired users' map exploration on a large tabletop: "Edge projection" (projection of the POIs onto the vertical and horizontal edges of the map), "Neighborhood Browsing" (separation of the map into several areas containing exactly one POI), and "Touch and Speak" (verbal directions from the current touch location to the POI).

Vibration-based guidance techniques

Tactons are structured, abstract vibro-tactile signals that convey information when modifying different parameters (frequency, amplitude, waveform, duration, and rhythm) [1]. Their advantage lies in the possibility to communicate information, even when ambient noise level is too high, when privacy should be guaranteed or when auditory environmental cues should not be masked. The latter is especially interesting for visually impaired people who use to rely on audio stimuli for many tasks. Different studies have used tactons for whole-body guidance, and may still hold for hand guidance. A first approach consisted in displaying tactons onto different body locations indicating the direction to follow. It has, for instance, been used with a tactile belt and spatialized vibrations on the chest [13]. A second approach used different vibration patterns that indicate directions. For example, the Tactile Compass [8] used one vibrational motor to indicate directions on a 360° circle by varying the length of two subsequent pulses within one pattern. Third, NaviRadar [11] used a single vibrator on a mobile device. When the user made a circular swipe on a touch screen, vibrational feedback was used to confirm the direction of travel.

Wearables for visually impaired people

Wearable devices have emerged in the past years and are now used in different contexts. For instance bracelets, such as the Jawbone Up, or smartwatches allow measuring heart rhythm, number of paces or sleep rhythm with the aim to contribute to a healthier life. Aside from the intended usage, they enable the design of new interaction concepts. These devices can have different forms, being integrated into clothing, gloves, glasses, jewelry, etc. [13]. The interest of wearable bracelets for visually impaired people has recently been demonstrated [16].

Quick-Glance and In-Depth Exploration

Approach overview

In order to explore a new map, we designed a two-step exploration technique including vocal and vibration feedback. The first exploration movement, called Quick-Glance in reference to the corresponding visual process, is performed above the map in mid-air. It supports the creation of a sparse but usable mental representation of the map. Interestingly, relying on mid-air movements, visually impaired people can explore a map without being stopped by physical objects located on the map. Thus, this method is compatible with tangible interactions on the map surface. The second step, called In-Depth, allows the user to select a POI in a given zone, and then reach it (Figure 1).

For the interaction between the user and the map, we used a smartwatch (Figure 3). This device presents multiple advantages. For instance, Profita et al. [10] showed in a recent study that wearing a device on the wrist was more acceptable than on any other body part. A smartwatch also contains a vibration mode that can easily be used for hand guidance. Furthermore, we



Figure 3. Smartwatch with infra-red markers

believe that in the near future, these devices will be largely adopted by visually impaired users.

Quick-Glance exploration

By skimming over the map with the smartwatch, users gain access to the names of the different zones as well as the number of POIs that they contain. We partitioned the map in twelve areas (Figure 2). These zones were arranged in a 3x4 grid, which was inspired by Kane's Neighborhood Browsing [5] with the difference of the grid being rectangular in our case. We suggest that this layout is easy to memorize because it corresponds to a T12 layout of a mobile phone keyboard. Indeed it has been observed, that visually impaired users appreciate the use of well-known keyboard layouts [6]. In our map, each zone could hold one or several POIs.



Figure 2. Map of France partitioned in 12 areas arranged in a 3x4 grid as in our prototype.

When moving the watch over an area, the number of POI that it contains is announced. In addition, POIs contained in this area are sent to the smartwatch. A

succession of finger swipes on the smartwatch touchscreen reads out the names of these different POIs. When users are interested in knowing the exact location of a given POI, they tap on the screen and the system switches to the In-Depth guidance mode.

In-Depth hand guidance

Once a POI is selected, the hand is guided towards the corresponding location with vibrations on the smartwatch. We chose this feedback in order to avoid annoying users with too much vocal information [3].

In our design, we had to guide the hand on small distances within a given area. Hence precision was important. We designed hand guidance following Rümelin's work on NaviRadar [11]. The NaviRadar technique provides information on directions and not distances. In a previous study on guidance with vibrational feedback, it has been shown that users preferred directional over distance information [8]. Figure 4 illustrates the method that we designed. Users perform clockwise circle with the finger on the screen. The smartwatch vibrates when the angle corresponds to the direction towards the selected POI. Users then move the finger in this direction. When the POI is reached, the smartwatch vibrates. At any moment, if they are not sure about the direction to follow, users can perform a new circle.

Prototype development

The Quick-Glance exploration technique requires knowing the position of the hand above the map. We used infrared tracking and placed markers around the map and on the smartwatch (Figure 3). We used a Sony Smartwatch 2 connected via Bluetooth to an Android smartphone. The smartphone was necessary

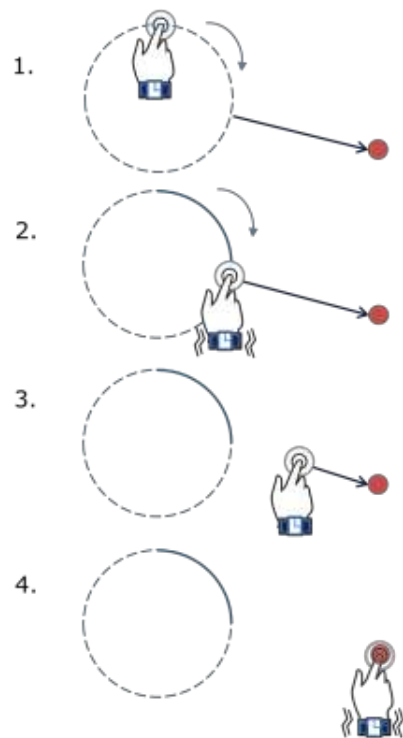


Figure 4. In-Depth finger guidance technique: 1) the user draws a circle clockwise; 2) when reaching the intersection with the line, the smartwatch vibrates; 3) the user moves his finger towards the target; and 4) the smartwatch vibrates as the finger reaches the target.

for executing the application and also for the audio output (the smartwatch provided only vibrational feedback). Our aim was to display the map on a large diffused illumination multi-touch table. However, the built-in infrared projectors underneath the surface interfered with the infrared tracking of the smartwatch. Thus in the current prototype, we used a tactile 22" screen (3M M2256PW). We plan to overcome this limitation by changing the position of the infrared cameras or using a multi-touch tabletop that is not based on infrared technology.

Discussion

Mid-air tracking

In order to create the final version of our technique, we will have to detect mid-air position of the hand. Two types of approaches can be used: inside-out or outside-in. Inside-out corresponds to the smartwatch being able to locate its own position above the table. This could be performed with magnetic sensors. Outside-in corresponds to using an external tracking of the hand. This could be done by using depth sensors such as Kinect or Leap Motion.

Detection of the fingertip position in mid-air

One problem of the current implementation is that infrared markers are attached to the smartwatch. Thus the wrist is actually located and serves as the point of reference. This is probably not as intuitive as the fingertips. Indeed, it could be confusing and disturbing for mental mapping. On the other hand, we suppose that a learning effect would occur after some training. Another solution could rely on a system attached to a finger. This solution might however be more cumbersome for the user as wearing a device on the fingertip may hinder exploratory movements. Yet,

these hypotheses have to be evaluated in future studies.

Conclusions and Future work

In this paper we presented a non-visual interaction method for improving the accessibility of geographic maps on large tabletops for visually impaired users. This interaction method consists in two steps: a "Quick-Glance" step to quickly acquire an overall knowledge of the map; and an "In-Depth" step to finely guide the hand towards a selected POI. The prototype has been implemented with a touchscreen and a smartwatch with auditory and tactile feedback.

Future work will consist in evaluating this prototype, in order to determine usability and accessibility for visually impaired people. This evaluation will be done in collaboration with the Institute for Blind Youth (CESDV-IJA) in Toulouse. Concerning the "Quick-Glance" technique we aim to evaluate whether it allows visually impaired users to quickly acquire usable knowledge about the map. Time needed for exploration and the quality of spatial learning will also be evaluated. It is indeed necessary to evaluate whether the T12 layout enables the creation of a usable mental representation. It will also be interesting to study whether this technique results in a higher cognitive load or physical fatigue (resulting from holding the arm in mid-air). Furthermore, as mentioned above, it will be necessary to investigate whether the tracking of the wrist instead of the finger impacts the map exploration process. Concerning the "In-Depth" technique we aim to explore the design space for hand guidance techniques. In addition to improving accessibility of maps on large surfaces to visually impaired people, we believe that

this work may contribute to the design space of spatial non-visual interaction.

Acknowledgments

We thank Julie Ducasse for helpful assistance regarding the design and Dr. Eric Barboni who mentored the internship of the first author (Master 1 Human-Computer Interaction at University of Toulouse, France) during which this work has been developed.

References

- [1] Brewster, S. and Brown, L.M. Tactons: structured tactile messages for non-visual information display. Proc. of AUIC'04, Australian Computer Society, Inc. (2004), 15–23.
- [2] Brock, A.M., Oriola, B., Truillet, P., Jouffrais, C., and Picard, D. Map design for visually impaired people: past, present, and future research. *Médiation et information - Handicap et communication* 36, 36 (2013), 117–129.
- [3] Brock, A.M., Truillet, P., Oriola, B., Picard, D., and Jouffrais, C. Interactivity Improves Usability of Geographic Maps for Visually Impaired People. *Human-Computer Interaction*, (to appear 2014).
- [4] Buxton, W. Multi-Touch Systems that I Have Known and Loved. 2007.
- [5] Kane, S.K., Morris, M.R., Perkins, A.Z., Wigdor, D., Ladner, R.E., and Wobbrock, J.O. Access Overlays: Improving Non-Visual Access to Large Touch Screens for Blind Users. Proc. of UIST '11, ACM Press (2011), 273–282.
- [6] Kane, S.K., Wobbrock, J.O., and Ladner, R.E. Usable gestures for blind people. Proc. of CHI '11, ACM Press (2011), 413–422.
- [7] Miele, J. a., Landau, S., and Gilden, D. Talking TMAP: Automated generation of audio-tactile maps using Smith-Kettlewell's TMAP software. *British Journal of Visual Impairment* 24, 2 (2006), 93–100.
- [8] Pielot, M., Poppinga, B., Heuten, W., and Boll, S. A tactile compass for eyes-free pedestrian navigation. *Interact 2011, LNCS 6947*, Springer Berlin Heidelberg (2011), 640–656.
- [9] Poppinga, B., Magnusson, C., Pielot, M., and Rasmus-Gröhn, K. TouchOver map: Audio-Tactile Exploration of Interactive Maps. Proc. of MobileHCI '11, ACM Press (2011), 545–550.
- [10] Profita, H.P., Clawson, J., Gilliland, S., et al. Don't mind me touching my wrist. Proc. of ISWC '13, ACM Press (2013), 89–96.
- [11] Rümelin, S., Rukzio, E., and Hardy, R. NaviRadar: a novel tactile information display for pedestrian navigation. Proc. of UIST '11, ACM Press (2011), 293.
- [12] Tatham, A.F. The design of tactile maps: theoretical and practical considerations. Proc. of international cartographic association: mapping the nations, ICA (1991), 157–166.
- [13] Tsukada, K. and Yasumura, M. ActiveBelt: Belt-Type Wearable Tactile Display for Directional Navigation. *UbiComp 2004, LNCS Vol 3205*, Springer Berlin Heidelberg (2004), 384–399.
- [14] Wang, Z., Li, B., Hedgpeth, T., and Haven, T. Instant Tactile-Audio Map: Enabling Access to Digital Maps for People with Visual Impairment. Proc. of Assets '09, ACM Press (2009), 43–50.
- [15] Yatani, K., Banovic, N., and Truong, K. SpaceSense: representing geographical information to visually impaired people using spatial tactile feedback. Proc. of CHI '12, ACM Press (2012), 415 – 424.
- [16] Ye, H., Malu, M., Oh, U., and Findlater, L. Current and future mobile and wearable device use by people with visual impairments. Proc. of CHI '14, ACM Press (2014), 3123–3132.

BCI hybride pour la palliation du handicap moteur sévère

Alban Duprès
LAGIS, Université Lille 1
Villeneuve d'Ascq, FRANCE
alban.dupres@univ-lille1.fr

José Rouillard
LIFL, Université Lille 1
Villeneuve d'Ascq, FRANCE
jose.rouillard@univ-lille1.fr

François Cabestaing
LAGIS, Université Lille 1
Villeneuve d'Ascq, FRANCE
francois.cabestaing@univ-lille1.fr

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670466>

Résumé

Cet article présente les travaux en cours concernant la mise en place d'une interface cerveau-ordinateur hybride (hBCI, hybrid Brain Computer-Interface). Notre objectif applicatif est la palliation du handicap moteur sévère chez des patients atteints de myopathie de Duchenne (DMD, Dystrophie Musculaire de Duchenne). Une hBCI consiste à associer différentes modalités de contrôle à une interface cerveau-ordinateur (BCI) pure. Dans notre cas nous associons une BCI basée sur l'imagination motrice à des modalités de contrôle basées sur l'EMG (Electromyographie) et sur des capteurs de mouvements distaux. L'idée est de pouvoir détecter un mouvement volontaire du doigt à trois niveaux différents de la commande motrice : cérébral, moteur et distal. L'information provenant de ces différents niveaux est fusionnée en tenant compte de l'état du patient afin de s'adapter au mieux au caractère évolutif de sa maladie et à sa forte fatigabilité au cours de la journée.

Mots-clés

BCI hybride, Myopathie de Duchenne, Fusion de données.

Mot clés de la classification ACM

H.5.m.

Human factors, Information interfaces and presentation (e.g., HCI) : Miscellaneous.

Introduction

Les interfaces cerveau-ordinateur (BCI) permettent une communication directe entre le cerveau d'un utilisateur et un dispositif externe, sans passer par l'intermédiaire des muscles et des nerfs périphériques. Pour l'instant, les BCI sont principalement destinées à la palliation du handicap moteur sévère. Dans ce cadre, d'autres technologies sont utilisées : suivi du regard [9] ou de la tête, interface exploitant l'électromyographie (EMG) [8] ou l'électro-oculographie, boule de commande (ou trackball), joystick, contacteurs, système de reconnaissance de la parole... Toutes ces technologies nécessitent, contrairement aux BCI, une commande musculaire. Les travaux relatés dans la littérature indiquent que ces techniques permettent, pour l'instant, un contrôle plus fiable et plus rapide que l'approche BCI [6]. Il est donc logique de privilégier ces dispositifs d'assistance lorsque la commande musculaire reste suffisamment précise pour les contrôler. Les BCI sont davantage réservées à des patients lourdement handicapés, n'ayant plus d'activité musculaire ou une activité erratique, pour lesquels ce type d'interface constitue l'unique moyen de communication.

Chez des patients atteints de DMD, la capacité de motricité évolue dans le temps, car elle est liée au caractère dégénératif de la maladie et à la fatigue du sujet au cours de la journée. C'est pourquoi il est intéressant, pour ces sujets, d'adapter l'outil d'assistance en leur proposant une BCI hybride qui associe à une BCI pure différentes modalités de contrôle. Plusieurs interfaces de ce type ont déjà été proposées : BCI et EMG [1], BCI et suivi du regard [7]... Nous focalisons nos travaux sur une interface hybride qui combine BCI, EMG et capteurs physiques. Les informations issues de ces différentes modalités sont fusionnées et pondérées en fonction de la fatigue du sujet pour répondre au mieux à l'évolution de

la commande motrice.

La première section décrit les trois modalités d'entrée enregistrées par notre système, la deuxième le traitement des données multimodales. La dernière section détaille le contrôle de trajectoire pour naviguer en environnement virtuel ou pour déplacer un curseur orienté sur un écran.

BCI hybride

Notre BCI hybride combine un système BCI avec de l'EMG et des capteurs physiques. Ce système permet de détecter l'exécution réelle ou l'intention d'exécution d'un mouvement — selon la fatigue du sujet — à différents niveaux de la commande motrice :

- Distal : grâce à des capteurs physiques qui détectent un mouvement de doigt ;
- Musculaire : grâce à l'EMG. Des électrodes placées à la surface de la peau enregistrent l'activité électrique des cellules musculaires. L'analyse du signal recueilli par ces électrodes permet de détecter des bouffées EMG caractéristiques de la contraction musculaire (voir figure 1) ;
- Cérébral : grâce à une BCI détectant une activité de type imagerie motrice.

La réalisation ou l'imagination d'un mouvement active la zone du cortex moteur dédiée aux muscles qui permettent ce mouvement. Cette activation est caractérisée par une modification, synchrone au mouvement, de la puissance des rythmes mu [8-12 Hertz] et bêta [13-26 Hertz]. On détecte ces rythmes dans l'EEG¹ enregistré sur le cuir chevelu de l'individu grâce à un casque permettant de recueillir le signal cérébral en positionnant des électrodes à l'aplomb des zones de projection sur le cortex moteur des muscles activés.

1. Electro-Encéphalo-Gramme

La modification du rythme mu est caractérisée par une diminution de sa puissance avant et pendant le mouvement. On parle de désynchronisation liée à l'événement (ERD, Event Related Desynchronisation). Après le mouvement, la puissance du rythme mu croît pour revenir à sa valeur de base, on parle alors de synchronisation liée à l'événement (ERS, Event Related Synchronisation) (voir figure 1). La modification du rythme bêta lors d'un mouvement suit une évolution similaire à celle du rythme mu. On observe une désynchronisation avant et pendant le mouvement suivie d'une synchronisation après la fin de ce dernier. La synchronisation du rythme bêta est davantage prononcée et caractérisée par le « rebond » bêta (voir figure 1).

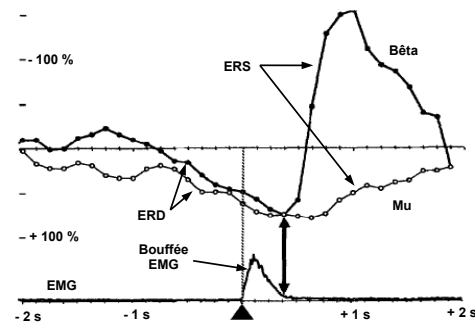


Figure 1 – Evolution des rythmes mu, bêta et de l'EMG lors d'un mouvement. Le triangle sur l'axe des abscisses marque le début du mouvement (source [3]).

Un paradigme de type imagination motrice (IM) permet de compléter l'information recueillie au niveau musculaire et distal. De plus, dans le cadre d'une utilisation prolongée, un paradigme de type IM est mieux adapté contrairement à un paradigme de type SSVEP

(Steady-State Visual Evoked Potentials) qui devient vite contraignant pour l'utilisateur [5]. Une fois que les trois modalités de contrôle sont mises en place, les données qu'elles fournissent sont traitées et fusionnées sur un logiciel adéquat pour permettre le contrôle en temps-réel de l'application.

Des systèmes incluant plusieurs modalités de contrôle ont été proposés dans la littérature, [10, 11]. Néanmoins ils diffèrent du notre sur plusieurs aspects. Le système proposé par [10] n'inclut pas de fusion des données. Il envisage uniquement la sélection d'une modalité de contrôle (joystick ou BCI) en fonction de mesures de qualité élaborées pour chacune d'entre elles. Dans [11], les auteurs fusionnent plusieurs canaux pour contrôler l'interface hybride. Cependant, ce système n'utilise pas de capteurs de l'activité distale. D'autre part, les canaux de contrôle sont fusionnés de façon statique, les poids de chaque modalité n'évoluant pas en fonction de l'état du patient. Notre travail s'inscrit dans la continuité de ces approches dans la mesure où nous envisageons l'utilisation d'une fusion dynamique qui permettra de prendre en compte les évolutions à court terme (fatigue) et long terme (dégénérescence) de l'état du patient.

Traitement des données multimodales

Dans le domaine de la multimodalité, la fusion ainsi que la fusion de données sont des mécanismes importants à mettre en œuvre en IHM. Nous nous intéressons dans nos travaux à la fusion en entrée, pour un système multimodal intégrant au moins trois sources de données [12]. On a donc besoin de fusionner l'information provenant de l'activité cérébrale de l'utilisateur que l'on tente de mesurer, renforcée par des données plus sûres comme l'activité musculaire et la détection des changements d'états des capteurs physiques.

Nous supposons que l'utilisateur peut réaliser — simultanément ou non — deux mouvements simples d'un doigt de la main gauche et d'un doigt de la main droite. L'amplitude, la force ou encore la cadence de ces mouvements — c'est-à-dire une information analogique associée à une latéralité — est mesurée par les trois modalités, EEG, EMG et capteur de mouvement (par exemple un micro-joystick). On dispose ainsi de 6 signaux analogiques qu'il faut combiner afin de détecter des mouvements ou des intentions de mouvement gauche ou droit. Après l'étape de fusion, nous disposons de deux signaux de contrôle dont les valeurs indiquent si l'utilisateur réalise ou imagine un mouvement gauche (signal $l(t)$) ou droit (signal $r(t)$).

L'architecture actuelle repose sur deux PC : le premier recueille et traite les données provenant de l'utilisateur et des capteurs (input), tandis que le second est chargé des sorties du système vers l'utilisateur (output). Nous utilisons le logiciel OpenVibe développé par INRIA (<http://openvibe.inria.fr/>) pour traiter, fusionner les données et prendre des décisions. Les communications entre les deux machines sur le réseau se font grâce au protocole VRPN (Virtual-Reality Peripheral Network).

Dans un premier temps, l'application proposée à l'utilisateur lui permet de faire évoluer un personnage dans un monde 3D pour lui apprendre à utiliser le système de façon ludique. Une application de type contrôle d'un curseur est envisagée dans un second temps.

Dans la version actuelle du système, l'utilisateur peut actionner, avec ses index, deux boutons poussoirs positionnés à sa gauche et à sa droite (voir figure 2). Ces boutons sont reliés à une carte Arduino, qui enregistre en continu leur état. Ces deux informations sont envoyées vers le logiciel OpenVibe de la même façon que les signaux

électrophysiologiques enregistrés par des électrodes placées sur l'utilisateur. Dans le plus simple des cas, lorsque les deux boutons sont actionnés simultanément, la fusion de données produit une trajectoire de l'avatar en ligne droite à vitesse constante. Dans les autres cas, les calculs de trajectoires sont dépendants du mode de pilotage paramétré que nous décrivons dans la section suivante.

Contrôle de trajectoire et curseur orienté

Ci-dessous, nous décrivons le contrôle de trajectoire d'un mobile dans un environnement virtuel, ou — ce qui est équivalent — de position et d'orientation d'un curseur dans un plan, à partir des signaux résultant de la fusion des trois modalités. En entrée du module de contrôle de trajectoire, nous disposons de deux signaux analogiques à valeurs positives, $l(t)$ et $r(t)$. Pour simplifier les notations, nous considérons que ces signaux sont continus — en pratique ils sont échantillonnés à la cadence imposée par le système d'acquisition — et nous omettons l'argument (t) .

Équations cinématiques

En additionnant et soustrayant l et r , nous élaborons deux signaux $s = r + l$ et $d = r - l$ qui permettent de contrôler respectivement la vitesse et le changement de direction du mobile. Le principe est simple : quand l et r sont égaux — c'est-à-dire quand le signal différence d est nul — le mobile se déplace en ligne droite. En revanche, quand le signal d est non nul, il contrôle la variation de l'orientation. Dans les deux cas, le signal somme s contrôle la vitesse du mobile. Ce mode de contrôle de trajectoire correspond à un mobile de type unicycle [4]. Les équations cinématiques permettant de calculer sa position et son orientation — définies par le vecteur d'état $(x \ y \ \theta)$ — sont données par :

$$\dot{x} = \alpha s \cos(\theta), \quad \dot{y} = \alpha s \sin(\theta) \quad \text{et} \quad \dot{\theta} = \beta d, \quad (1)$$



Figure 2 – Notre prototype d'interface BCI hybride

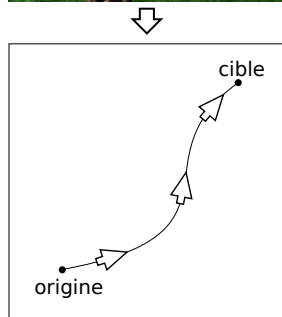
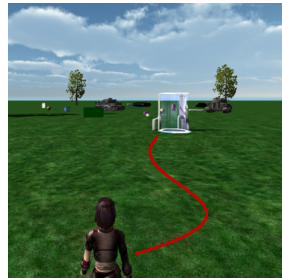


Figure 3 – Navigation virtuelle vs. curseur 2D + orientation

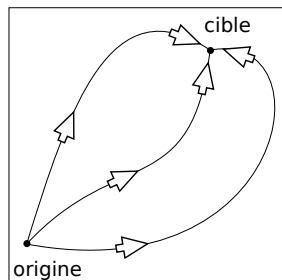


Figure 4 – Différentes trajectoires pour une même cible

dans lesquelles α et β sont des gains autorisant une adaptation de la cinématique au comportement spécifique d'un utilisateur. Le signe de β permet notamment de fixer la préférence vis-à-vis du changement d'orientation, pour un pilotage de type « char d'assaut » quand β est positif (rotation vers la gauche quand r est supérieur à l) ou un pilotage de type « guidon de vélo » quand β est négatif (rotation vers la gauche quand l est supérieur à r).

En pratique, il est nécessaire d'insérer des fonctions de transfert non-linéaires dans ces équations cinématiques pour faciliter la commande de trajectoire. En premier lieu, afin de pouvoir suivre une trajectoire parfaitement rectiligne quand les signaux l et r ne sont pas égaux mais simplement « proches » l'un de l'autre, on peut ramener à zéro le signal d quand sa valeur absolue est inférieure à un seuil ϵ . On peut également appliquer une transformation non linéaire au signal s afin d'autoriser des accélérations et décélérations rapides du mobile, comme cela est réalisé par la plupart des systèmes d'exploitations pour le contrôle du déplacement du curseur d'un dispositif de pointage [2].

Curseur orienté

Le vecteur d'état dont l'évolution est régie par les équations cinématiques (1) inclut le degré de liberté θ correspondant à l'orientation du mobile. En utilisant les mêmes équations cinématiques, nous proposons de piloter le déplacement d'un curseur orienté sur une surface plane. Quand il maîtrise l'application de navigation, pour passer en mode de contrôle d'un curseur orienté, l'utilisateur peut s'imaginer qu'il observe maintenant le déplacement du mobile depuis un point de vue fixe situé à l'aplomb de la scène. Le passage du mode navigation au mode curseur orienté est illustré par la figure 3.

Le principal défaut de ce mode de contrôle d'un curseur est qu'il s'avère assez compliqué de « revenir en arrière »

lorsqu'on a dépassé la cible qu'on souhaitait atteindre avec un degré de précision élevé. En effet, pour réaliser ce retour en arrière, il faut tout d'abord maintenir l'une des commandes à une valeur élevée jusqu'à ce que le mobile ait fait demi-tour, puis avancer dans la direction opposée de celle qui avait permis initialement de s'approcher de la cible. En revanche, si on suppose que l'utilisateur dispose d'une bonne maîtrise du contrôle de trajectoire, il peut atteindre une cible en suivant des trajectoires différentes à partir d'un même point origine (cf. figure 4).

Ainsi, une fois la cible atteinte, on dispose de l'orientation finale du curseur comme variable complémentaire à la position. Cela permet par exemple de modifier l'action réalisée au travers de l'interface lors de la sélection. Pour des utilisateurs en situation de handicap moteur sévère, comme les personnes atteintes par la myopathie de Duchenne, disposer de ce degré de liberté supplémentaire est d'une importance majeure.

Conclusion

Cet article présente la mise en place d'une BCI hybride combinant plusieurs modalités de contrôle : BCI, EMG et capteurs physiques. Ce système est conçu pour détecter le mouvement d'un doigt à différents niveaux de la commande motrice. En tenant compte différemment de chaque composante, le système peut s'adapter à la motricité changeante de l'utilisateur. Il s'agit de travaux en cours dans la mesure où le système n'est pas encore pleinement fonctionnel. Concernant la composante BCI, la désynchronisation du rythme mu reste délicate à mettre en évidence de façon fiable. La synchronisation du rythme bêta (rebond bêta) est davantage visible et peut être utilisée dans un premier temps pour détecter un mouvement. Néanmoins, étant donné que le rebond bêta n'est observable qu'après le mouvement de l'utilisateur, un

certain laps de temps est nécessaire entre la réalisation du mouvement et sa détection effective. Concernant la fusion des données, aucun modèle n'a été déterminé même si dans un premier temps une approche possibiliste est envisagée. Enfin, l'évaluation de la fatigue musculaire chez le patient reste également problématique. Plusieurs indicateurs de fatigue musculaire peuvent être utilisés : indices extraits des signaux EMG, difficulté à réaliser la tâche en cours, etc.

Références

- [1] Barreto, A. B., Scargle, S. D., Adjouadi, M., et al. A practical EMG-based human-computer interface for users with motor disabilities. *Journal of Rehabilitation Research and Development* 37, 1 (2000), 53–64.
- [2] Casiez, G., and Roussel, N. No more bricolage ! methods and tools to characterize, replicate and compare pointing transfer functions. Dans *Actes d'UIST'11, 24th ACM Symposium on User Interface Software and Technology* (2011), 603–614.
- [3] Derambure, P., Defebvre, L., Bourriez, J.-L., Cassim, F., and Guieu, J.-D. Désynchronisation et synchronisation liées à l'événement. étude de la réactivité des rythmes électrocorticaux en relation avec la planification et l'exécution du mouvement volontaire. *Neurophysiologie Clinique/Clinical Neurophysiology* 29, 1 (1999), 53–70.
- [4] Dudek, G., and Jenkin, M. *Computational Principles of Mobile Robotics*. Cambridge University Press, Cambridge CB2 2RU, 2000.
- [5] Duprès, A., Cabestaing, F., and Rouillard, J. BCI exploitant les SSVEP : étude de la stabilité du classifieur dans le temps et des effets de l'apprentissage humain sur les performances de classification. Dans *Actes de la conférence Handicap'2014* (2014), 105–112.
- [6] Edlinger, G., Kapeller, C., Espinosa, A., Torrellas, S., Miralles, F., and Guger, C. Multi-modal computer interaction for communication and control using EEG, EMG, EOG and motion sensors. Dans *Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion*. Springer, 2013, 633–641.
- [7] Finke, A., Koesling, H., and Ritter, H. Multi-modal human-computer interfaces for handicapped user groups : Integrating mind and gaze. Dans *Actes de Conf. Int. ACM on Ubiquitous Computing (UbiComp2010)* (2010), 1–4.
- [8] Jeong, H., and Choi, J. An EMG-controlled graphic interface considering wearability. Dans *Actes de Human Computer Interaction, INTERACT'03* (2003), 958–961.
- [9] Kosmyna, N., and Tarpin-Bernard, F. Une combinaison de paradigmes d'interaction cerveau-ordinateur et suivi du regard pour des interactions multimodales. Dans *Actes de la conférence Ergonomie, Interaction Homme-Machine (Ergo'IHM)* (2012), online.
- [10] Kreilinger, A., Kaiser, V., Breitwieser, C., Williamson, J., Neuper, C., and Müller-Putz, G. R. Switching between manual control and brain-computer interface using long term and short term quality measures. *Frontiers in neuroscience* 5 (2011).
- [11] Leeb, R., Sagha, H., Chavarriaga, R., and del R Millán, J. A hybrid brain-computer interface based on the fusion of electroencephalographic and electromyographic activities. *Journal of neural engineering* 8, 2 (2011), 025011.
- [12] Tarby, J.-C., and Rouillard, J. BCI et IHM : premiers retours d'expériences. Dans *Actes d'ERGOIA 2010* (2010).

Predictive Usability Evaluation: Aligning HCI and Software Engineering Practices

Káthia Marçal de Oliveira

UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
kathia.oliveira@univ-
valenciennes.fr

Sophie Lepreux

UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
sophie.lepreux@univ-
valenciennes.fr

Christophe Kolski

UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
christophe.kolski@univ-
valenciennes.fr

Ahmed Seffah

UVHC, LAMIH CNRS UMR 8201
59313 Valenciennes, France
ahmed.seffah @univ-
valenciennes.fr
Lappeenranta University of
Technology (LUT)
P.O Box 20 FI-53851
Lappeenranta, Finland
ahmed.seffah@lut.fr

Abstract

Can we - software developers, usability experts, user interface designers - predict usability from the early user interface (UI) design artifacts and models? Can we define predictive measures to evaluate usability without a concrete UI? These questions seemed natural for us since UI modeling (task, user, concepts, etc.) is being largely explored in recent years for the automatic generation of final UI. To answer those questions we propose a model-based predictive usability evaluation approach that uses a set of usability measures. These measures are the essence of a framework we are developing for usability prediction. Initial empirical studies were performed to support this approach. This paper presents the fundamental basis on top of which we have developed this approach.

Keywords

Usability, measures, evaluation

Keywords – ACM classification

H.5.2. Information interfaces and presentation (e.g., HCI): User Interface – Evaluation/Methodology.

Introduction

During the last two decades, usability evaluation has been intensively treated by Human-Computer Interaction (HCI) researchers and usability professionals. A large number of usability evaluation

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670467>

methods and models have been proposed. An exhaustive account of these methods can be found in [4]. Several models and standards ([2], [3]) have been established to measure usability as a quality construct. Mainly these models break down usability into measurable attributes. They also provide a list of measures that are generally in conjunction with usability evaluation methods to qualitatively and quantitatively estimate usability attributes. Building on or contributing to these standards and usability evaluation methods, several European projects suggested specific frameworks to measure usability; representative examples include SUMI (Software Usability Measurement Inventory) and MUSiC (Metrics for Usability Standards in Computing) developed by the Human Factors Research Group, University College, Cork. Trying to integrate different models in a same structure from a literature review, Seffah et al. [11] proposed a consolidated model, named QUIM (Quality in Use Integrated Measurement). QUIM is a unifying model that assesses usability by using 12 factors, 29 criteria and more than 120 measures. Another significant research is the Single Usability Measure suggested to standardize usability measure into a single global score [9]. Moreover, guidelines and ergonomic criteria for usability evaluation were also proposed (e.g. [10]).

However, most of those projects do not address explicitly usability prediction from early UI design models. This is because the calculation of the proposed measures in these projects requires a fully functional prototype. Among the few initiatives, one can mention, the European action COST294 that highlighted the need for theoretical frameworks defining the nature of interaction quality. This includes techniques and

measures for the evaluation of the interaction quality during design. Another initiative is the proposition of Panach et al. [7] to represent usability features related to functionality abstractly in a conceptual model, and then to carry out automatically their implementation. To that end, they proposed a method to define the functional usability features (e.g. feedback support, user input error prevention, go back) in a model-driven approach using what they call conceptual primitives. This method includes the identification of usability mechanisms, changes in conceptual models and in the compilers. Measures are also proposed for early usability evaluation considering these conceptual primitives. To interpret the result of the measures, they assign categorical values (e.g. very good, good, bad) for range of numerical values based on guidelines in the literature. However, the classification of measure results into categories still remains intuitive and is not correlated with the end user view. Considering this scenery, we plan to work on a definition of a framework for predictive usability evaluation from early design models based on measures that can be interpreted using the perception of the users about the usability of final user interfaces. No extra effort is required in the UI design to allow the quality evaluation.

Next sections present our approach to perform predictive usability evaluation, a preliminary investigation on this direction and some final remarks.

Definition of an Approach for Predictive Usability Evaluation

The context we envision is that UI designers can, during UI modeling and early design phases, apply the specific measures related to that model and have some indication of the level of usability of the final UI using

pre-defined correlations. The new UI being developed and evaluated can later also have its measures integrated in the historical base of evaluations, thus improving the correlations.

To that end, we have planned to define a framework that consists of a set of measures applicable to different UI design models from a high abstraction level to the UI implementation [6]. The prediction of usability and the analysis of measures results are based on the possible correlations between predictive usability measures (on the design artifacts) and the results of usability tests performed by users (on the final UI). The correlations will be defined with several empirical studies.

Figure 1 portrays the proposed predictive usability prediction approach as well as the way it can supplement inspection-based evaluations. Predictive usability evaluations are performed on various design models using usability predictive measures.

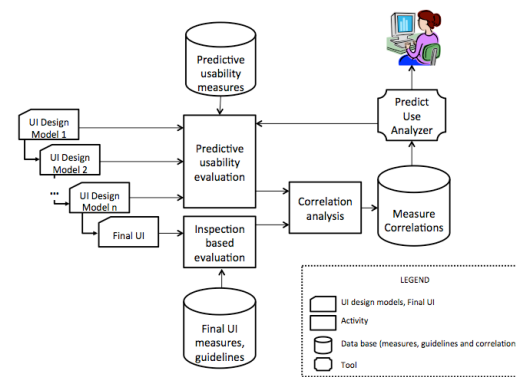


Figure 1. Predictive Usability Evaluation Approach

The predictive usability evaluation uses a database including the correlations gathered from previous empirical experiments. Each experiment defines and refines continuously the correlations between the results of usability measures collected from early UI design models and the results of inspection-based evaluations made on the final UI. A predictive usability analyzer tool (PredictUse Analyzer) will support the measures collection and calculation as well as the usability prediction that uses the database.

A preliminary investigation

We had surveyed different quality models that propose a list of measures. Our starting point was the QUIM model [11] since it is defined considering several quality models and it organizes a set of usability measures. While investigating those measures, we noted that most of the measures were applicable only to final UI since the main goal of QUIM is the definition of quality in use that means, while using the UI. We decided therefore to analyze also the internal measures of usability (applied to a non-executable software product) from ISO9126. The ISO9126 external measures has already been included in QUIM. Finally, we also analyzed the ergonomic criteria from the most cited usability model, defined by [10].

Our procedure was first to read the measure definition (description, formula and scales when available). Next, considering the features of each design model, we identified if that measure could be applied to that design model even that the formula customization would be needed. Finally, we analyzed the measures against the three UI design models proposed in the CAMELEON framework [1]:

- Task model - that describes the various users' tasks to be carried out with a UI. Concur Task Tree (CTT) [8] is the most used notation to define this model;
- The abstract User Interface - that defines the rendering of the task model independently from any style of interaction (e.g. graphical interaction, vocal interaction, speech synthesis and recognition) as well as the capability and constraints of the platform;
- The concrete user interface – that concretizes an abstract UI into an interactor-dependent expression for a given context of use;
- Final UI – that is the operational user interface running in a specific platform.

To do the analysis, we took into account the features of each one of the UI models. For instance, the task model defined using CTT, is composed basically of a hierarchy of different kind of tasks (abstract, interaction, user and application tasks) and temporal relationships among tasks (interleaving, synchronization, enabling, enabling with information passing, choice, deactivation, iteration and optional).

Table 1 presents some measures extracted from QUIM. We discovered that some measures require the execution of the UI (e.g. temporal efficiency), therefore, it will be applied to the final UI only. Others are related to the UI design itself (e.g. local density), are thus related to the concrete UI and some of them to abstract UI. Finally, some measures are related to the information to be presented in the UI (e.g. interface shallowness), thus they can be applied in all UI design models from the highest level of abstraction (the task model). After analyzing measurable criteria

(considering QUIM measures, SQUARE standard, and Scapin and Bastien criteria [10]), we started the customization of them to evaluate each one of the models.

Table 1. Some examples of the analysis of measures

Measures	Task model	Abstract UI	Concrete UI	Final UI
Function understandability	✓		✓	✓
Human Efficiency	✓		✓	✓
Interface Appearance Customizability			✓	✓
Interface shallowness	✓	✓	✓	✓
Layout Appropriateness		✓	✓	✓
Local Density		✓	✓	✓
Longest Depth	✓	✓	✓	✓

Table 2 presents the analyses of some measures applicable to task model. We note that we have qualitative and quantitative measures. The qualitative measures are collected by inspections with UI designers. Those inspections are cognitive walkthrough based [5], since the assessment is done by experts who should wonder the potential perception of the user in the use of UI implemented based on that task model. The quantitative measures can be calculated from the data obtained from task models tools (e.g. CTTE).

It is important to highlight that each one of those measures is not complete, that means, each measure is defined to measure what can be measured in the task model considering only the elements of this model. However, in other models and in the final UI other aspects of the same measure should be evaluated. For example, the measure defined for information density considers only the aspect number of input/output in the same task since this task will probably be designed in

the same panel/window (if we consider graphical UI). However, when evaluating this measure in an abstract UI or in a concrete UI we can look for the organization of those input/output related to the whole space of the panel/window.

Table 2. Examples of measure for task model

Measure	Measurement
Function understandability	Subjective evaluation with five point ordinal scale (1-5)
Longest Depth	Computes the number of nodes in the CTT hierarchy from the root till the primitive task
Visual Coherence	$100 - \text{Number of primitive task that are not closely related in the same high level task} / \text{number of primitive tasks defined} * 100$
Minimal Actions	Compute the minimum path of access to primitive task
Input validity data	$(\text{Number of sequence of "Enabling with information passing"} + \text{number of tasks explicitly defined to validate the data}) / \text{Number of tasks is decomposed in at least one primitive task}$

We also highlight that some transversal features cannot be evaluated in a task model since they are usually not modeled. These transversal features include: help, interface customization aspects, specific aspects of navigability such as home, exit, etc.

Finally, we note that our idea is not to define an interpretation of each measure. This will be done by comparing the result of evaluation for a specific project with other results plotted in a graph of correlation points. This graph correlates measures of models with the evaluation of final UI.

To allow the prediction evaluation we need to compare these results with historical data of previous evaluations where correlations were discovered between the evaluation of the task models and the final UI. The definition of a database of correlation takes time since we need several UI designs of several projects, the definition of the UI based on those models and the evaluation from UI design models and final UI.

However, to evaluate the feasibility of our idea, we performed a case study with students of the second year of the two-year graduate course on computer science at the University of Valenciennes. Although we had little data (collected from 14 projects done by students during the course) for an adequate predictive evaluation, we would like to show an example of the proposed approach. Figure 2 presents a graph that shows the evaluation of the task model (x axis) for the measure input validity data and the evaluation of the same measure in a final UI (y axis) (we used a scale from 1 (input never validated) to 5 (input always validated)). It shows that when the task model has a good evaluation the final UI tends to have a better evaluation too. We obtained a value of Spearman correlation equals to 0.46, which is not very good, but that we considered acceptable since we do not have a lot of points (14 projects) and the subjects (i.e., the students) had very little experience in task modeling and programming.

This preliminary study showed us that there a lot of work need to be perform to effectively validate this approach since we must have UI design models and the final UI developed based on those models.

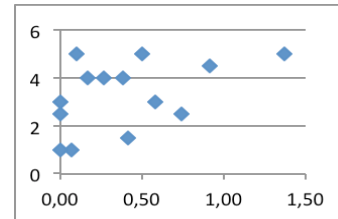


Figure 2. Evaluation of UI design model (x axis) x Final UI for input validity data (y axis)

Conclusion and Future Work

This paper presents the first results of a long-term research that aims to define a predictive usability

References

- [1] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers*, v. 15, n. 3 (Jun 2003), 289–308.
- [2] ISO 9241-11. Ergonomic requirements for office work with visual display terminals - Part 11: Guidance on Usability (1998).
- [3] ISO/IEC 25023. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality (Aug 2011).
- [4] Ivory, M. Y., Hearst, M. A. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, v. 33, n. 4 (Dec 2001), 470-516.
- [5] Mahatody, T., Sagar, M., Kolski, C. State of the Art on the Cognitive Walkthrough method, its variants and evolutions. *International Journal of Human-Computer Interaction*, 26 (8)(2010), 741-785.
- [6] Oliveira, K. New Research Challenges for User Interface Quality Evaluation. *Actes d'IHM 2010*, (Sep 2010), 145-148.

framework to be used in UI design. The main advantage of this approach is that the interpretation of the quality evaluation of the UI design models by measures is supported by previous evaluations of the end user in similar projects. The main limitation of this approach is the need of previous correlation analysis to define the prediction of usability. To have good correlations, a large number of projects is needed to evaluate the design models and the final UI. We are now working on the definition of the database of correlations with controlled projects in the computer sciences courses. Future works include the complete definition of the framework and its application in large projects.

- [7] Panach, J. I., Juzgado, N. J., Pastor, O. Including functional usability features in a model-driven development method. *Computer Science and Information System Journal*, v.10, n.3 (2013), 999-1024.
- [8] Paterno, F., Mancini, C., Meniconi, S. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models, *INTERACT '97* (Jul 1997), 362-369.
- [9] Sauro, J., Kindlund E. A method to standardize usability metrics into a single score. *CHI'2005* (2005), 401-409.
- [10] Scapin, D.L. and Bastien, J.M.C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information Technology*, v. 16, n.4 (1997), 220-231.
- [11] Seffah, A., Donyaee, M., Kline, R. Padda, H. Usability measurement and measures: A consolidated model, *Software Quality Journal*, v. 14 (2006), 159-178.

Challenges for Usability Testing in Ubiquitous Systems

Carla Bezerra

Federal University of Ceará
Fortaleza – CE, Brazil
carlailane@great.ufc.br

Rossana M. C. Andrade

Federal University of Ceará
Fortaleza – CE, Brazil
rossana@great.ufc.br

Rainara Maia Santos

Federal University of Ceará
Fortaleza – CE, Brazil
rainarasantos@great.ufc.br

Mourad Abed

UVHC, LAMIH, CNRS UMR 8201
mourad.abed@univ-valenciennes.fr

Káthia Marçal de Oliveira

UVHC, LAMIH, CNRS UMR 8201
kathia.oliveira@univ-valenciennes.fr

José Maria Monteiro

Federal University of Ceará
Fortaleza – CE, Brazil
monteiro@lia.ufc.br

Ismayle Santos

Federal University of Ceará
Fortaleza – CE, Brazil
ismaylesantos@great.ufc.br

Houcine Ezzedine

UVHC, LAMIH, CNRS UMR 8201
houcine.ezzedine@univ-valenciennes.fr

Abstract

Ubiquitous computing expands both the place where the software system is used and the traditional way of interacting with its users. This happens since technologies should be fully integrated in the user daily activities in such way they become indistinguishable. So, in this scenario, new characteristics can emerge like calmness, transparency and context awareness that should be evaluated to assure the ubiquitous system quality. One of the methods used for evaluation of traditional systems is usability testing. Then, in this paper, we identify the main challenges to perform usability testing in ubiquitous systems based on an overview of the literature studies and on our own experience in testing this kind of systems. We also describe our ongoing work where we investigate usability testing to be designed considering context-awareness and to be supported by specific measurements to the ubiquitous computing scenario.

Keywords

Usability Testing, Ubiquitous Systems, Quality.

Keywords ACM classification

H.5.m. Information interfaces and presentation (e.g., HCI) : Miscellaneous. H.5.2 User Interfaces. (D.2.2, H.1.2, I.3.6): Evaluation/methodology.

Introduction

Ubiquitous computing is a new paradigm that makes technological devices that interact with users in the environment invisible to them, shifting their attention

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.

<http://dx.doi.org/10.1145/2670444.2670468>

from the software/hardware to everyday tasks [1]. Ubiquitous systems have features of distributed, heterogeneous and user-centered systems. A challenge for these systems is thus to design, implement and test these features as well as to attend issues related to user satisfaction [2].

According to Nielsen [3], usability describes the ease in which users interact with a given interface correctly using system features. Usability testing is an observation method that aims to evaluate interactive systems from the observation of the use of the systems by users [4]. It is then explicitly devoted to analyze in details how users interact with the application for accomplishing well-defined tasks. Typical data collected during user testing are user execution time, number of errors, and user satisfaction. After the test completion, the collected data are then interpreted and used to improve the level of the application usability [5].

Nelly et al. [6] state that the evaluation in ubiquitous systems from the viewpoint of Human-Computer Interaction (HCI) could be difficult, since the evaluation of HCI approaches tend to be subjective and/or fragmented. From the viewpoint of HCI, one of the best and most used methods for evaluating ubiquitous systems is usability testing (e.g., see [7, 8, 9]).

After a literature review, we have observed the implementation of usability testing in ubiquitous systems is generally performed in the same manner as in traditional systems. However, we have also identified several gaps and challenges of applying usability testing in ubiquitous systems that need more investigation and we gather them to present in this paper based on this review and on our own experience.

Besides that, we present our ongoing research to deal with some of these gaps and challenges.

Literature Review

We have performed a non-systematic review of the literature to identify different studies in the literature that proposed evaluation of ubiquitous systems. In this review, 20 papers were analyzed taken into account the following criteria: application domain, measures of usability testing, usability testing approaches in ubiquitous systems, and environment to performing of usability testing (controlled or real).

Regarding the domain of the evaluated ubiquitous applications, they have been diversified. However, most papers performed usability evaluation in Smart Homes [2, 10, 11, 12, 13] and Navigating System [14, 15, 16, 17, 18]. Other work performed usability evaluation in other domains [7, 9, 19, 20, 21, 22] or not really performed any evaluation [8, 23, 24, 25].

In the literature review, several measures were identified to assess the usability of the applications. However, we found out that no paper identifies measures for usability testing that consider the characteristics of ubiquitous applications such as context information. Most of the identified measures are the same used in traditional usability testing systems, for example: List of Tasks Performed [9, 13, 14, 15, 18, 20], Learnability [15, 20, 22, 25]; Response time [2, 14, 16, 25] and User Satisfaction [7, 11, 13, 14]. Only three papers [8, 19, 24] have identified measures that consider the characteristics of ubiquitous systems. Other papers did not provide measures for usability testing [7, 10, 18].

Most papers [2, 7, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21, 22] performed usability testing in controlled environments, not in real environments. Moreover, few papers presented specific approaches to usability evaluation for ubiquitous applications and, these approaches that have been identified are not specific to usability tests. In fact, we just found out evaluation frameworks [8, 10], usability evaluation processes [11] and measures of usability evaluation in ubiquitous systems [2, 19].

Experience Report

We have been working with HCI evaluations in ubiquitous systems using software quality measurement. To do that, we have first defined which HCI quality characteristics of ubiquitous systems should be measured. Using a literature systematic mapping, we have identified the following five characteristics specific for evaluating HCI in ubiquitous systems: Context-Awareness (corresponds to the ability to collect contextual information and use them to make adaptations), Transparency (system proactivity, so the user is minimally distracted), Attention (refers to continuous or uninterrupted use of the systems as the user moves through several devices), Mobility (represents an increase of the attention's focus of the user in their everyday activities) and Calmness (prevents humans from feeling overwhelmed by technology). For these characteristics, a set of software measures were defined to evaluate the HCI quality. Examples of such measures are to context-awareness: Adaptation Correctness, Context Correctness and Adaptation Time [26].

The defined measures were collected in evaluations of three ubiquitous applications to mobile devices. The

first application is called UbiPrinter, which aims to support the printing of documents by finding the user nearest printer. The second one is called UbiMute that monitor the user calendar looking for events in which the user cannot answer calls, for example: "meeting". Then, this application switched to the "silent" mode of the user's mobile phone. The last application is called GREat Tour, which is a tour guide for the GREat¹ laboratory. It provides information about the environments of the laboratory that the user is visiting.

To execute the measure collection, we use different methods to collect all necessary data to calculate the defined measures: (i) generation of user interaction logs with the application by the code instrumentation; (ii) application of questionnaires to the final user as well as to the application developer; and (iii) observation form to the evaluator who makes notes while the user is using the application.

We faced some difficulties along the preparation and execution of these tests:

- Each application requires a specific implementation code to collect the necessary data for the evaluation. For instance, to evaluate Adaptation Correctness, different context information had to be collected for each application: in UbiPrinter, we should automatically collect the Wi-Fi; and in UbiMute, the calendar.
- Each application requires different test procedures. For instance, to observe the behavior of the UbiMute in the real environment, it is required that

¹ GREat - Group of Computer Networks, Software Engineering and Systems.

the application should be used during at least some days in the own users mobile phones. As a consequence, the participation of an observer during the test was not possible. On the other hand, we could assess UbiPrinter and Great Tour with an observer watching users of these applications.

- And, evaluating context-awareness requires the execution of several tests with a detailed planning of all tasks to be followed and the setting of different context information. For instance, to test the context-awareness of Great Tour, three mobile devices were necessary, each one with different level of battery, because this application has a battery level based adaptation.

Challenges

Based on the literature review about usability testing and in our own experience in HCI quality evaluation of ubiquitous systems, challenges in performing usability testing of these systems have been identified.

An identified key challenge is related to the environment of usability testing. Most usability testing is performed in the laboratory. However, according to Kim et al. [8], the traditional controlled laboratory tests are not suitable for ubiquitous applications. This is due to the large number of resources demanded to the environment required by ubiquitous systems, which must be the same used in real environment [12]. So, usability testing in ubiquitous systems should be suitable for running in real environments.

Another challenge identified is that ubiquitous environments have more usability factors that should be used, such as contextual variables [10]. A context-

aware application has part of its particular behavior dependent of the context [27]; therefore, it is necessary to predict all relevant changes in context and analyze when those changes can impact the behavior of the system application [28]. Usability testing should be mobile, modular and contextual to meet the ubiquitous systems features, and should also employ user-driven tasks to be accurate and relevant [7]. Test cases should be designed considering the context that the user can have in the environments in which the application will run. It is also important to consider the calm interaction related to the user's attention factor. This is a feature that shows the transparency of ubiquitous application in relation to user interaction.

For measures used in usability testing, we found out that most of these measures does not consider the factors of ubiquitous applications. It is a challenge to make the evaluation of the usability of these systems more reliable, and identify measures that consider the ubiquitous features in usability tests [19].

Finally, another challenge is in relation to the implementation of usability testing in ubiquitous systems, because currently they follow the same activities performed in traditional systems. There were not identified in our review one specific approach for usability testing of ubiquitous systems. Thus, research can be conducted to elaborate an approach for usability testing with specific tasks and specific measures to evaluate ubiquitous systems.

Future Work

Based on the challenges identified in usability testing for evaluating ubiquitous systems, we are working on a research project to try to solve them. The project objective is to define a strategy for planning and

executing context-aware testing for ubiquitous systems, considering the context, and using measurements.

To that end, we should evolve usability testing to evaluate the complete ubiquitous system. We advocate that the interaction issues in ubiquitous systems go beyond the human-computer relationship, encompassing the interaction between different devices and systems, which we call in this project actor-computer interaction, in which the actor can be not only the user, but also, another computer, device or software system in order to better answer the need of invisibility required.

Références

- [1] Maia, M. E., Fonteles, A., Neto, B., Gadelha, R., Viana, W., & Andrade, R. M. C. LOCCAM-loosely coupled context acquisition middleware. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 534-541 (2013, March).
- [2] Lee, J., & Yun, M. H. Usability assessment for ubiquitous services: Quantification of the interactivity in inter-personal services. In Management of Innovation and Technology (ICMIT), 2012 IEEE International Conference on (pp. 718-724), (2012, June).
- [3] Nielsen, J. Usability Engineering. Boston: Academic Press (1993).
- [4] Rubin, J., & Chisnell, D. Handbook of usability testing: how to plan, design, and conduct effective tests. John Wiley & Sons. Nielsen, J. Usability Engineering". Boston: Academic Press (2008).
- [5] Fiotakis, G., Raptis, D., Avouris, N. Considering Cost in Usability Evaluation of Mobile Applications: Who, Where and When", INTERACT '09 Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I (2009).

Based on this project, we intend to build a specific approach to usability testing for ubiquitous systems. To reach this objective, this project includes: (i) Establish a software process for context-awareness testing based on evidence; (ii) Define measures to evaluate the interoperability of ubiquitous systems; (iii) Design context-awareness test cases based on measurements; and (iv) Develop and experimentally evaluate support tools for collecting measures.

ACKNOWLEDGMENTS

This work is a partial result of the Maximum project supported by FUNCAP (FAPs/INRIA/INS2i-CNRS 11/2011) and Cactus project supported by CNPQ (Universal 14/2013).

- [6] Neely, S.; Stevenson, G.; Kray, C.; Mulder, I.; Connelly, K. and siek, K.A. Evaluating pervasive and ubiquitous systems," IEEE Pervasive Computing, Vol. 7, no. 3 pp. 85-8 (2008).
- [7] Riihiahho, S. User testing when test tasks are not appropriate. In European Conference on Cognitive Ergonomics: Designing beyond the Product Understanding Activity and User Experience in Ubiquitous Environments (p. 21), (2009, September).
- [8] Kim, H. J., Choi, J. K., & Ji, Y. Usability evaluation framework for ubiquitous computing device. In Convergence and Hybrid Information Technology. ICCIT'08,(Vol. 1, pp. 164-170), (2008, November).
- [9] Silveira, C., Eloy, L., & Monteiro, J. M. A query language for data access in ubiquitous environments. Proceedings CILEI electronic journal, (2010).
- [10] Kim, S. H., Kim, S. W., & Park, H. M. Usability challenges in ubicomp environment. Proc. of Int'l. Ergonomics Assoc. (2003).
- [11] Lee, J., Song, J., Kim, H., Choi, J., & Yun, M. H. A user-centered approach for ubiquitous service

evaluation: an evaluation metrics focused on human-system interaction capability. In *Computer-Human Interaction* (pp. 21-29), (2008, January).

[12] Acampora, G., Loia, V., & Vitiello, A. Using ANFIS and FML for deploying transparent services in smart environments. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, Sixth International Conference on (pp. 628-633), (2012).

[13] Song, I. J., & Cho, S. B. Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment. *Expert Systems with Applications*, 40(5), 1827-1838, (2013).

[14] Ross, T., & Burnett, G. Evaluating the human-machine interface to vehicle navigation systems as an example of ubiquitous computing. *International Journal of Human-Computer Studies*, 55(4), 661-674 (2001).

[15] Poláček, O., Grill, T., & Tscheligi, M. Towards a navigation system for blind people: a wizard of oz study. *ACM SIGACCESS Accessibility and Computing*, (104), 12-29, (2012).

[16] Noguera, J. M., Segura, R. J., Ogáyar, C. J., & Joan-Arinyo, R. A scalable architecture for 3D map navigation on mobile devices. *Personal and ubiquitous computing*, 17(7), 1487-1502, (2013).

[17] Maly, I., Mikovec, Z., Vystrcil, J., Franc, J., & Slavik, P. An evaluation tool for research of user behavior in a realistic mobile environment. *Personal and ubiquitous computing*, 17(1), 3-14, (2013).

[18] Rodriguez-Sanchez, M. C., Moreno-Alvarez, M. A., Martin, E., Borromeo, S., & Hernandez-Tamames, J. A. Accessible smartphones for blind users: A case study for a wayfinding system. *Expert Systems with Applications*. *Journal Expert Systems with Applications*, (2014).

[19] Scholtz, J., & Consolvo, S. Toward a framework for evaluating ubiquitous computing applications. *Pervasive Computing*, IEEE, 3(2), 82-88 (2004).

[20] McGlenn, K., O'Neill, E., Gibney, A., O'Sullivan, D., & Lewis, D. SimCon: A Tool to Support Rapid Evaluation of Smart Building Application Design using Context Simulation and Virtual Reality. *J. UCS*, (2010).

[21] Gavalas, D., & Kenteris, M. A web-based pervasive recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing*, (2011).

[22] Gavalas, D., & Kenteris, M. Evaluation of a web recommender system in electronic and mobile tourism. *International Journal of Web Engineering and Technology*, 7(1), 4-21, (2012).

[23] Ranganathan, A., Al-Muhtadi, J., Biehl, J., Ziebart, B., Campbell, R. H., & Bailey, B. Towards a pervasive computing benchmark. In *Pervasive Computing and Communications Workshops*. *PerCom 2005 Workshops*, (pp. 194-198), (2005, March).

[24] Rubio, J. M. L., & Bozo, J. P. Approach to a Quality Process for the Ubiquitous Software Development. In *Electronics, Robotics and Automotive Mechanics Conference*, (pp. 701-705), (2007).

[25] Tang, L., Yu, Z., Zhou, X., Wang, H., & Becker, C. Supporting rapid design and evaluation of pervasive applications: challenges and solutions. *Personal and ubiquitous computing*, 253-269, (2011).

[26] Santos, R. M., de Oliveira, K. M., Andrade, R. M., Santos, I. S., & Lima, E. R. A Quality Model for Human-Computer Interaction Evaluation in Ubiquitous Systems. In *Human Computer Interaction* (pp. 63-70). Springer International Publishing (2013).

[27] Tse, T. H.; Yau, S. S.; Chan, W. K.; Lu, H.; Chen, T. Y. Testing Context-Sensitive Middleware-Based Software Applications. In *28th Annual International Computer Software and Applications Conference*, Washington, p. 458-466 (2004).

[28] Wang, Z.; Elbaum, S.; Rosenblum, D. S. (2007). Automated Generation of Context-Aware Tests. In *Proceedings of the 29th international conference on Software Engineering (ICSE '07)*, p. 406-41.

Interfaces cerveau-ordinateur et jeux sérieux : adaptation aux patients schizophrènes

Piau Charlotte

LAGIS, Lille 1
FRANCE
charlotte.piau@outlook.fr

Bekaert Marie-Hélène

LAGIS, Lille 1
FRANCE
marie-helene.bekaert@univ-lille1.fr

Cabestaing François

LAGIS, Lille 1
FRANCE
fancois.cabestaing@univ-lille1.fr

Résumé

Les interfaces cerveau-ordinateur (BCI) peuvent être utilisées comme outil de rééducation pour un sujet en situation de handicap. Leur utilisation dans le cadre d'un handicap moteur a déjà fait ses preuves. Nous proposons ici de les utiliser dans le cadre d'un handicap psychique : la schizophrénie, au travers de jeux sérieux pour la rééducation des troubles psychiques associés. Nous soulignons l'importance d'adapter les BCI utilisées au handicap mais aussi l'importance d'impliquer le patient dès le début du développement des interfaces afin de maximiser ses performances et ainsi faciliter sa réadaptation.

Mots clés choisis par les auteurs

BCI, Handicap, Schizophrenia, Adaptation, Feedbacks, Serious Game

Mot clés de la classification ACM

H.5.m. Human factors, Information interfaces and presentation (e.g., HCI): Miscellaneous.

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670469>

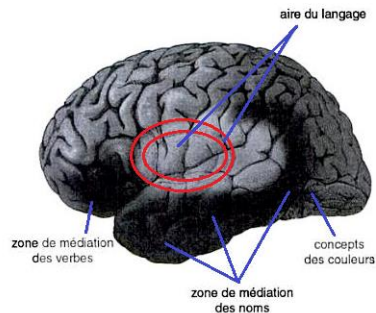


Figure 1. Schéma du cerveau avec localisation en rouge les aires du langage [15].

Introduction

Les interfaces cerveau-ordinateur sont des interfaces homme-machine (IHM) permettant une communication directe entre le cerveau d'un utilisateur et un dispositif externe, tel un bras articulé, un ordinateur ou un fauteuil roulant [21]. L'activité électrique cérébrale est mesurée puis traitée et traduite en commande ce qui permet de piloter le dispositif, sans passer par les canaux usuels que sont les nerfs et les muscles. Dans le cas d'un handicap psychique, les BCI peuvent être utilisées comme outil de rééducation et/ou de réadaptation en complément des traitements médicamenteux.

Dans cet article nous proposons de définir l'adaptation des BCI au patient schizophrène (SCZ) et à son handicap. En première partie nous traitons l'adaptation de l'interface au handicap psychique puis nous verrons l'importance de l'implication du patient.

BCI : adaptation au handicap psychique.

Les troubles psychiques résultent de nombreuses pathologies. Ils peuvent être traités par les médicaments et/ou par la rééducation. Avec la démocratisation de l'outil informatique et notamment avec l'essor des smartphones et tablettes, les jeux sérieux se sont développés comme outil de rééducation ou de réadaptation. Dans le cas de l'autisme, par exemple, ils sont utilisés pour améliorer les compétences linguistiques des patients [2].

Dans notre équipe nous nous focalisons entre autre sur les troubles mentaux liés à la schizophrénie [14] et plus particulièrement les hallucinations verbales auditives (AVH). L'implication des BCI dans le traitement des troubles psychiques est relativement récente et fortement liée à la stimulation cérébrale profonde. Par exemple, elles sont utilisées dans ce cadre, pour les troubles obsessionnels

compulsifs (TOC) ou celui du déficit d'attention avec ou sans hyperactivité [3]. Très peu d'articles traitent de l'utilisation des BCI non invasives pour le traitement des troubles psychiques et plus particulièrement pour ceux liés à la schizophrénie [11].

L'intérêt de l'utilisation du neurofeedback dans ces pathologies est « l'effet de transfert », c'est-à-dire la capacité du patient à employer les stratégies apprises lors des sessions d'entraînement pour réduire ou augmenter son activité neuronale dans une zone déterminée et ce en absence de neurofeedback. A ce jour, des études [1] [16] [18] ont utilisé le neurofeedback chez une population SCZ qui a été capable, par la suite, d'autoréguler son activité neuronale en dehors de tout feedback.

BCI et jeux sérieux pour la gestion des Hallucinations Verbales Auditives

La schizophrénie est la psychose la plus répandue chez l'adulte. Dans 70% des cas, les sujets SCZ souffrent d'AVH qui dans 25 à 30% des cas, sont résistantes à toutes formes médicamenteuses [17].

Les sujets SCZ sont caractérisés par des profils électrophysiologiques particuliers [19]. L'amplitude de leurs ondes thêta et gamma sont par exemple réduites par rapport à un sujet contrôle. Les ondes thêta (4-7hz) sont à la fois impliquées dans l'orientation spatiale et dans la relaxation, tandis que les ondes gamma (30 – 200 Hz) sont impliquées dans les fonctions cognitives hautes, telle que la mémorisation ou le calcul.

Nous proposons d'utiliser les BCI pour aider les patients SCZ souffrant d'AVH. Notre but final est d'entraîner le patient via des jeux sérieux, en dehors de ses périodes de crises. L'entraînement devrait permettre à celui-ci, grâce à l'« effet

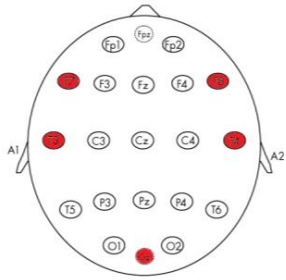


Figure 2. Système 10-20 pour le placement des électrodes. En rouge celles utilisées [8].

transfert », de mettre en œuvre les stratégies apprises à l'entraînement pour réguler son activité neuronale et ainsi influencer sur la fréquence de survenue des AVH ou sur leur sévérité. Pour cela différentes applications BCI, développées via le logiciel Unity [20], sont proposées. Quatre axes de développement sont mis en œuvre :

- L'instrumentalisation du chant ;
- La maîtrise du niveau de relaxation ;
- L'orientation spatiale ;
- L'association consigne-images.

Pour chacun d'eux nous enregistrons les rythmes thêta et gamma en occipital, et au niveau du gyrus temporal supérieur (STG) et du gyrus frontal inférieur (IFG), c'est-à-dire sur l'électrode F7, F8, T3, T4 et Oz (système 10-20) [Figure 1 et 2]. Ces trois zones sont impliquées dans les AVH. En effet avant la survenue d'une AVH, on observe une désactivation des signaux électro physiologiques au niveau des aires du langage : STG et IFG.

Une étude de 1990 montre que les AVH seraient moins fréquentes lorsque le sujet fredonne [7]. L'instrumentalisation du chant dans un jeu sérieux permettrait au sujet SCZ d'apprendre des stratégies musicales applicables lorsqu'il ressent la survenue de sa crise afin d'éviter ou minimiser cette dernière.

La BCI destinée à la maîtrise du niveau de relaxation a pour but d'apprendre au sujet SCZ à mieux synchroniser ses ondes thêta, à l'aide d'un feedback auditif et/ou visuel. L'interface pourrait, par exemple, afficher comme retour visuel un cercle dont il faudrait augmenter par paliers la circonférence, laquelle serait corrélée à la synchronisation des ondes thêta. Une consigne auditive pourrait indiquer la rythmique

d'inspirations/expirations que le patient doit adopter pour faciliter sa relaxation.

La BCI basée sur l'orientation spatiale, dans un labyrinthe de complexité croissante (nombre d'embranchements croissant, présence ou absence d'impasses et de fausses pistes dans les niveaux supérieurs), est envisagée afin de stimuler les ondes thêta. Le sujet met en œuvre progressivement des stratégies d'orientation spatiale au travers d'une quête type chasse aux trésors. Plus le sujet SCZ améliore ses performances (augmentation de l'amplitude des ondes thêta) plus la complexité de navigation dans l'environnement s'accroît.

La dernière interface (association consigne-images) est basée sur une récente étude qui a démontré la possibilité d'augmenter volontairement les ondes gamma chez des individus sains [9] (Figure 3). Deux phases de jeux y sont proposées : soit la consigne est indépendante de la nature de l'image et de sa position à l'écran ; soit elle est dépendante de la nature de l'image, mais toujours indépendante de sa localisation. A chaque succès (réponse correcte à la consigne), un retour sonore est donné. L'objectif à atteindre est d'augmenter l'occurrence du signal auditif. Il ressort de cette étude qu'entre le premier essai et le dernier, la puissance du signal recueilli sur l'électrode Oz (Figure 2) a considérablement augmenté, principalement dans la bande 30-60 Hz. L'entraînement a donc permis d'augmenter la puissance et la synchronisation des ondes gamma. Une BCI basée sur le même principe d'association consigne-images pourrait proposer au patient SCZ un jeu évolutif (plusieurs niveaux de difficulté et divers scénarii) en fonction de ses performances. Le but étant d'amener progressivement le patient à augmenter son score (corrélé à la puissance du signal dans la bande gamma), à travers des bonus ou des challenges. Si des résultats similaires à l'étude citée étaient obtenus avec des

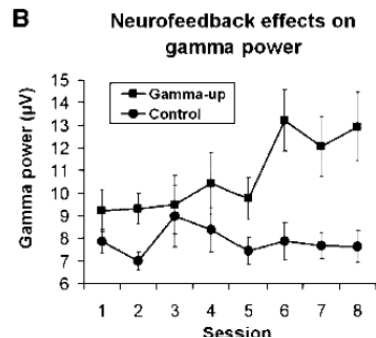


Figure 3. Effets du neurofeedback sur la puissance du signal dans la bande de fréquences gamma au sein du groupe « gamma-up » et du groupe contrôle. Au cours des 8 sessions, il est observé une augmentation de puissance pour la bande gamma (30-60 Hz) [9].

patients SCZ, nous pourrions alors regarder si un entraînement sur le long terme permettrait grâce à l'« effet de transfert » de réduire la sévérité des AVH par l'augmentation de la puissance et de la synchronie des ondes cérébrales dans la bande gamma, comme McCarthy en fait l'hypothèse dans son étude [11].

Adaptation de l'interface à la pathologie

Avant de développer une BCI pour la rééducation ou la réadaptation, il faut connaître et respecter les spécificités de la pathologie. C'est ainsi que pour des personnes épileptiques on évitera d'avoir recours au SSVEP et aux flashes [6].

Dans le cas de la schizophrénie, d'autres contraintes apparaissent. Lorsqu'on se réfère à des études antérieures [5], on voit qu'un sujet SCZ est sensible à certaines interférences visuelles. C'est ainsi qu'un objet animé dans son champ visuel périphérique le détournera de ce qui se situe au centre. Cette haute sensibilité ne concerne que les objets animés. En effet pour un objet interférant par coloration, le sujet SCZ possède une sensibilité équivalente à un sujet contrôlé [4].

De ce fait les applications développées doivent être les plus simplistes possibles, à la fois dans le graphisme, dans le nombre d'objets animés et leurs interactions et éviter les interférences. Par exemple, dans les applications de type labyrinthe classique, l'utilisateur peut se repérer grâce à un plan situé en périphérie où sa position est marquée par un symbole clignotant. Dans un jeu sérieux de type labyrinthe, ce symbole animé perturberait l'utilisateur SCZ et le détournerait de son objectif. Dans la mesure où c'est le mouvement qui entraîne la perturbation, nous proposons d'avoir recours à une coloration de la zone incluant sa position dans le labyrinthe lui-même.



Figure 4. Visuel de l'interface Hand-ball proposée par OpenVibe [13].

L'implication du patient, mise en œuvre d'une conception centrée sur l'utilisateur.

Les travaux de Nijboer [12], montrent que le succès d'une BCI dépend du niveau d'implication de l'utilisateur, c'est-à-dire de la motivation dont il va faire preuve pour interagir avec la machine. Or la motivation est dépendante de multiples paramètres subjectifs, de la personnalité ou de l'âge de l'utilisateur.

C'est ainsi qu'entre un sujet de 40 ans et un sujet de 20 ans l'affinité avec le visuel proposé va changer, de même pour deux sujets de même âge selon leurs goûts et centres d'intérêts.

Récemment notre équipe a mené une expérience avec un sujet sain, lycéen, sur l'utilisation de différentes interfaces cerveau-ordinateur proposées dans le logiciel Open-Vibe [13]. Le but de l'expérience était d'observer le ressenti d'un sujet jeune vis à vis du visuel et de comparer en terme de performance les images motrices choisies pour atteindre l'objectif. Il a été observé que pour les applications dont le design le décourageait (P300 speller), les performances obtenues étaient significativement basses et la phase d'entraînement anormalement longue.

Le logiciel Open-Vibe propose un jeu très simple (Figure 4) dont la commande est réalisée par imagerie motrice, grâce à des électrodes positionnées sur le cortex moteur. Lorsque le sujet imagine un mouvement de la main droite, classiquement un mouvement d'ouverture-fermeture de la main, le ballon se dirige du centre (position de repos) vers la cage de droite, et vice-versa. Quand ce même lycéen a été amené à utiliser le jeu de hand-ball, il s'est approprié l'application dès la première utilisation et a dépassé nos attentes en réussissant dès le premier essai à exécuter la tâche d'imagerie mentale motrice demandée.

On voit au travers de cette petite expérience à quel point la motivation est importante, mais aussi à quel point le retour perceptif proposé va permettre à l'utilisateur de s'immerger facilement ou non dans la tâche qui lui est demandée.

De même l'image motrice utilisée est à personnaliser en fonction de l'utilisateur. Dans le cadre du jeu de handball nous avons proposé à notre lycéen d'utiliser différentes imageries motrices pour voir si l'image choisie influe sur la performance. Certaines imageries motrices ont été proposées par l'équipe (dribbler, écrire, pianoter...), d'autres ont été choisies spontanément par le sujet (frapper à une porte, essorer une éponge, ...). Nous avons constaté que les performances obtenues variaient sensiblement en fonction de l'image choisie. La meilleure performance a été obtenue lors d'une tâche d'imagerie libre, le patient choisissant lui-même l'image motrice à exécuter, celle de l'éponge.

On peut supposer qu'à l'instar de notre lycéen, les jeunes sujets seront naturellement attirés vers les interfaces proches des jeux vidéo actuels. Alors que des sujets plus âgés préféreront des interfaces moins recherchées et plus simples dans leur design, des applications moins intuitives, plus rationnelles et plus monochromes.

Références

- [1] Bolea A.S., Neurofeedback treatment of chronic inpatient schizophrenia, *J Neurother*, 14 (2010), 47-54.
- [2] Carter C.M., Using Choice with Game Play to Increase Language Skills and Interactive Behaviors in Children with Autism, *Journal of Positive Behavior Interventions*, 3 (2001) 131-151.

Dans le cas des patients SCZ, et de nombreuses pathologies invalidantes, le sujet présente souvent une fatigabilité accrue et des déficits d'attention. Il est important de tenir compte de ces limites à la fois dans la durée des parties et dans la complexité de l'environnement présenté. Une étude réalisée en 2013 sur 61 sujets sains, montre par exemple que l'utilisation d'une BCI comme interface de jeu, nécessite un temps d'adaptation lié à la difficulté d'utilisation de cet outil [10].

Il est donc important lors du développement de l'interface de prendre en compte à la fois la pathologie visée mais aussi l'âge et la personnalité des utilisateurs, afin de garantir une implication haute et ainsi avoir une garantie de résultats.

Conclusion

Quel que soit le handicap que vise à pallier les BCI, il est important d'adapter ces interfaces à la pathologie du sujet mais aussi au sujet lui-même. C'est pourquoi dans notre future collaboration avec le CHR de Lille, le patient sera impliqué dès le début de l'étude afin d'adapter l'outil à sa pathologie. Ainsi le sujet se sentira impliqué et cette implication maximisera les scores de performance.

- [3] Clausen J., Bonding Brains to Machines: Ethical Implications of Electroceuticals for the Human Brain, *Neuroethics*, 6-3 (2013), 429-434.

- [4] Ducato M.G. et al. Attentional capture in schizophrenia: Failure to resist interference from motion signals, *Cognitive Neuropsychiatry*, 13 (2008), 185-209.

- [5] Ducato M.G. et al., Attentional capture in schizophrenia and schizotypy: Effect of attentional load, *Cognitive Neuropsychiatry*, 13 (2008), 89–111.
- [6] Fisher, R. S. et al., Photic and Pattern induced Seizures: A Review for the Epilepsy Foundation of America Working Group, *Epilepsia*, 46-9 (2005), 1426-1441.
- [7] Green M.F. and Kinsbourne M., Subvocal activity and auditory hallucinations: Clues for behavioral treatments?, *Schizophrenia Bulletin*, 16-4 (1990), 617.
- [8] Jasper H.H., Report of the committee on methods of clinical examination in electroencephalography: 1957, *Electroencephalography and Clinical Neurophysiology* 10-2 (1958), 370–375.
- [9] Keizer A.W. et al., The effect of gamma enhancing neurofeedback on the control of feature bindings and intelligence measures, *Int J Psychophysiol*, 75-1 (2010), 25-32.
- [10] Liarokapis, F. et al., Assessing Brain-Computer Interfaces for Controlling Serious Games. Dans *Actes VS-Games* (Sep. 2013).
- [11] McCarthy-Jones S., Taking Back the Brain: Could Neurofeedback Training Be Effective for Relieving Distressing Auditory Verbal Hallucinations in Patients With Schizophrenia?, *Schizophrenia Bulletin*, 38-4 (2012), 678–682.
- [12] Nijboer F. et al., Ethical, psychological and social implications of brain-computer interface application in paralyzed patients. Dans *AAI Workshop* (Sep. 2006).
- [13] Open-Vibe : <http://openvibe.inria.fr/>
- [14] Piau C. and Bekaert M-H., Utilisation des Interfaces Cerveau Ordinateur par des patients schizophrènes présentant des hallucinations verbales auditives. Dans *Handicap 2014* (June 2014).
- [15] Rondal J.A. and Seron X., Troubles du langage: bases théoriques, diagnostic et reeducation, *Edition Mardaga*, (1999), 19.
- [16] Ruiz S. et al. Acquired self-control of insula cortex modulates emotion recognition and brain network connectivity in schizophrenia, *Hum Brain Mapp.*, 34-1 (2013), 200-2012.
- [17] Shergill, S. S. et al., Auditory hallucinations: a review of psychological treatments. *Schizophrenia Research*, 32 (1998), 137–150.
- [18] Surmeli T, Ertem A, Eralp E, Kos IH. Schizophrenia and the efficacy of qEEG-guided neurofeedback treatment: a clinical case series, *Clinical EEG Neurosci*, 43-2 (2012), 133-144.
- [19] Uhlhaas P.J. and Singer W., Abnormal neural oscillations and synchrony in schizophrenia, *Nature Reviews Neuroscience*, 11-2 (2010), 100–113.
- [20] Unity <http://unity3d.com/>
- [21] Wolpaw, J. R. et al., Brain-computer interfaces for communication and control. *Clin Neurophysiol*, 113 (2002), 767–791.

Impact de l'utilisabilité, la motivation et l'expérience utilisateur sur l'engagement à utiliser une application mobile

Isabelle Tissier

CRP Henri Tudor
29 avenue J.F Kennedy
Luxembourg, 1855 Luxembourg-
Kirchberg
isabelle.tissier@tudor.lu

Guillaume Gronier

CRP Henri Tudor
29 avenue J.F Kennedy
Luxembourg, 1855 Luxembourg-
Kirchberg
Guillaume.gronier@tudor.lu

Résumé

Cette recherche s'intéresse aux manières d'engager des citoyens à utiliser durablement une application mobile. L'objectif est de déterminer la place de l'utilisabilité, de la motivation et de l'expérience utilisateur (UX) pour modifier et l'évaluer l'engagement.

Mots clés choisis par les auteurs

Engagement, expérience utilisateur, utilisabilité, application mobile, motivation

Mot clés de la classification ACM

H.5.m. Information interfaces and presentation (e.g., HCI): User Interfaces.

Introduction

Que ce soit dans le milieu éducatif, professionnel, ou privé, l'engagement des individus dans les actions qu'ils entreprennent est un élément crucial. À lui seul, il détermine la prise de décision, le commencement d'une activité ou le maintien d'un comportement ou d'une action. Appliquée aux interactions hommes-machines, l'importance de cette notion prend une toute autre dimension. Soit une technologie est engageante et elle sera fréquemment utilisée voir même intégrée aux

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670470>

habitudes de vie ; soit la technologie n'est pas engageante ce qui diminuera sa fréquence d'utilisation voire son rejet ou son abandon total. La création de technologie engageante devient donc décisive pour la survie d'une technologie [1], [2].

Les facteurs de l'engagement en IHM

Les premiers travaux sur l'engagement sont issus du champ de la psychologie sociale et font le lien entre l'engagement, les comportements et les attitudes. Ainsi, un individu est plus enclin à modifier ses attitudes et comportements sur le long-terme, lorsqu'il perçoit son engagement comme libre de toute pression et résultant de ses propres choix. L'engagement est alors vu comme un puissant facteur de changement librement consenti [3]. Avec l'avènement des technologies, les chercheurs en IHM tentent de définir l'engagement des utilisateurs de technologies. O'Brien & Toms [4] définissent l'engagement comme une qualité de l'expérience utilisateur (UX) qui dépend de plusieurs facteurs tels que l'attrait esthétique, la nouveauté, l'utilisabilité du système, la capacité de l'utilisateur à s'impliquer dans l'expérience, et l'évaluation de la prééminence globale de l'expérience par l'utilisateur [4].

L'engagement de l'utilisateur est un processus cyclique non linéaire (Figure 1). Un utilisateur s'engage à interagir avec une technologie de façon discontinue. Ainsi, le cycle de l'engagement se compose d'une étape qui est le point d'engagement, d'une période d'engagement effectif, puis d'une période de désengagement. Ensuite, le réengagement dépendra de tous les facteurs empiriques précités (attrait esthétique, nouveauté, utilisabilité, capacité à

s'impliquer, degré d'immersion, l'évaluation globale) (figure 1).

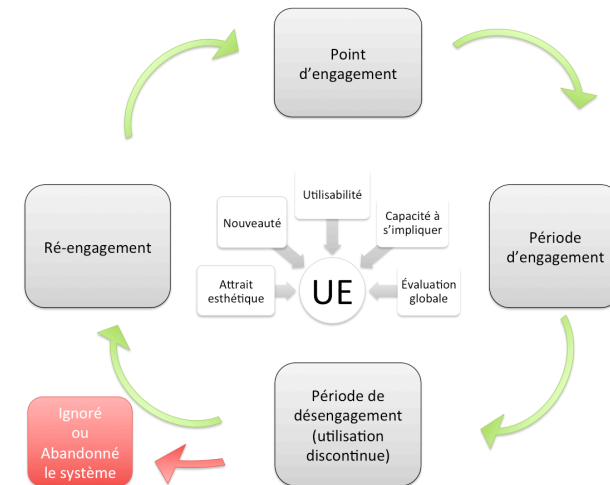


Figure 6. Processus de l'engagement [4]

Pour les besoins de notre recherche, nous avons choisi de nous intéresser à trois principaux facteurs dérivés du modèle de O'Brien et Toms [4]. Ainsi, nous avons cherché à mesurer la motivation d'utilisateurs vis-à-vis d'une application mobile, l'UX et l'utilisabilité. Chacun de ces facteurs est décrit dans les parties suivantes.

L'engagement et la motivation

La motivation correspond à la manifestation et à l'expression d'un engagement à travers une action. Dans le cadre des IHM, elle influence et détermine l'engagement d'un individu dans un comportement donné envers une technologie, et permet de maintenir ou non l'usage de cette technologie dans le temps [5].

Cette motivation est à considérer sous deux aspects majeurs [5]: intrinsèque et extrinsèque. La motivation intrinsèque, correspond aux activités réalisées pour la satisfaction inhérente qu'elles procurent sans attendre ni dépendre de rétributions ou de sanctions externes. La motivation extrinsèque représente les activités menées dans le but d'obtenir une récompense ou d'éviter une punition venant de l'extérieur.

L'engagement et l'utilisabilité

Nielsen [6] décrit l'utilisabilité à l'aide de 5 facteurs clés : la facilité d'apprentissage, la prévention des erreurs, la satisfaction d'utilisation, la facilité de mémorisation et enfin l'efficacité. Une technologie ne répondant pas à ces critères peut créer des problèmes d'utilisation qui peuvent mener les utilisateurs à se désengager vis-à-vis de celle-ci. L'utilisabilité est donc un critère complexe lié à l'expérience d'engagement [2], [7].

L'engagement et l'expérience utilisateur

L'UX est définie comme une évaluation bonne ou mauvaise, momentanée et principalement émotionnelle de l'interaction avec une technologie [8]. Cette UX est subjective, singulière, dynamique, temporelle et dépend du contexte ainsi que de l'expérience antérieure de l'utilisateur [8]. L'UX et l'engagement sont entremêlés et mutuellement tributaires l'un de l'autre. « Plus une expérience engage efficacement, plus elle sera mémorable » [9].

Problématique et démarche expérimentale

Dans le cadre d'un projet de recherche européen, nous avons souhaité engager des utilisateurs-citoyens à réaliser des relevés d'odeurs environnementales à l'aide d'une application mobile. Ces relevés devaient permettre de diminuer les nuisances olfactives, sources

de gêne et de désagrément pour les riverains d'une usine de fabrication de papier. Par conséquent, nous avons cherché à mesurer et à intégrer, au sein de l'application, des éléments motivationnels, d'utilisabilité et d'UX.

Cette étude s'est ainsi déroulée en deux étapes, dont seule la première est présentée dans ce papier. La seconde est en cours de réalisation :

- Dans une première étape, nous avons recueilli les éléments motivationnels des utilisateurs, ainsi que leurs opinions concernant l'utilisabilité et l'UX de l'application. Cette étape a servi de premier indicateur (en tant qu'interface de référence) sur le niveau d'engagement des utilisateurs, afin que ce niveau soit comparé avec l'étape suivante.
- Dans une seconde étape, nous avons cherché à intégrer de nouveaux éléments motivationnels, d'utilisabilité et d'UX au sein de l'application. Cette seconde version de l'application devait être comparée à la première en termes de qualité d'engagement. A ce jour, cette étape est en cours de réalisation.
- Expliquez les références locales (ex. : tout le monde ne connaît pas tous les noms de ville dans une région particulière).

Méthodologie

Un cas d'application : le projet OMNISCIENTIS

Le projet européen FP7 OMNISCIENTIS (Odour monitoring and information system based on citizen and technology innovative sensors - <http://www.omniscientis.eu/>) avait pour objectif de mieux gérer les nuisances olfactives émanant d'une entreprise de fabrication de papier. Pour ce faire, des

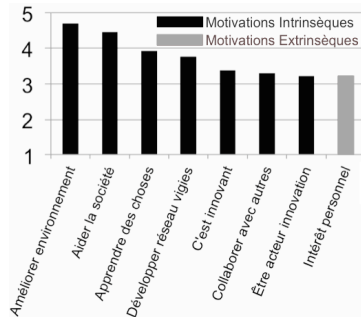


Figure 2. Motivation des vigies (1= pas du tout d'accord, 5= tout à fait d'accord).

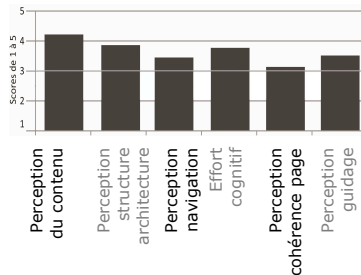


Figure 3. Résultat au DEEP [11] (1 = pas du tout, 5 = tout à fait).

citoyens, riverains de l'entreprise, étaient invités à réaliser des relevés d'odeurs à l'aide d'une application mobile disponible sur Android et Windows Phone. Ces relevés permettaient ainsi à l'entreprise de mieux comprendre les retombées olfactives de leurs procédés de fabrication, pour ainsi mieux les gérer et les réduire quand cela était possible. 31 citoyens, appelés vigies, se sont alors engagés volontairement à indiquer la présence ou l'absence d'odeur, son intensité (faible, moyenne ou forte), son type (pipi de chat, chou, soufre, etc.) et la gêne occasionnée (faible, moyenne ou forte). Sur ces 31 vigies, seules 13 d'entre elles sont très actives et réalisent des relevés quotidiens.

Mesures de la motivation, de l'utilisabilité et de l'UX
Pour recueillir, dans une première étape, les motivations des vigies, ainsi que leurs opinions concernant l'utilisabilité et l'UX de l'application mobile, nous leur avons transmis deux questionnaires en ligne. Le premier questionnaire portait sur les motivations, et obtint treize réponses. Le second questionnaire portait quant à lui sur l'expérience utilisateur et l'utilisabilité de l'application, et comptabilisa six participants. L'échantillon des répondants correspond à un public masculin peu technophile âgé en moyenne de 58 ans.

Le premier questionnaire se composait de dix-huit items interrogeant les vigies sur leur motivation à participer au projet [10]. Son objectif était de saisir les dimensions de l'application qui doivent être intégrées ou développées pour satisfaire les besoins motivationnels des vigies.

Le second questionnaire comportait trois échelles :

- le DEEP (Design-oriented Evaluation of Perceived usability) pour évaluer l'utilisabilité de l'application [11] ;
- l'AttrakDiff, qui permet d'obtenir un score d'UX indiquant la qualité de l'interaction des vigies avec l'application [12].
- le questionnaire d'acceptation technologique de Kim, Choi & Han qui mesure l'aspect global de l'application et l'intention de continuer à l'utiliser [13].

Ces questionnaires étaient complémentaires et permettaient de comprendre l'expérience et l'avis des vigies sur l'application. Ces mêmes questionnaires devaient être répliqués pour évaluer la nouvelle version de l'application, intégrant des éléments motivationnels, d'utilisabilité et d'UX

Premiers résultats

Les premiers résultats sont issus des réponses des vigies aux différents questionnaires de l'étape une.

Le questionnaire sur la motivation met en évidence l'aspect intrinsèque des motivations des vigies. Ainsi, leurs principales motivations sont de contribuer à l'amélioration de leur environnement, d'aider la société, d'apprendre des choses et de développer le réseau de vigies (Figure 2). Les scores résultants du DEEP (Figure 3) [11] et du questionnaire de Kim, Choi & Han (Figure 4) [13] sont tous au-dessus du niveau moyen (3). Cela révèle un niveau d'utilisabilité et un aspect général satisfaisant. De la même façon, les résultats de l'AttrakDiff [12] montrent une UX standard (Figure 5) et positive (entre 0,8 et 1,4).

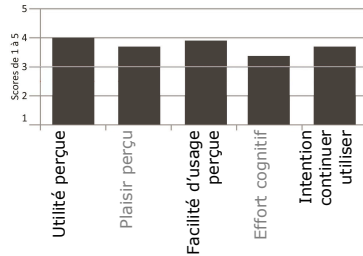


Figure 4. Résultat questionnaire de Kim, Choi et Han [13] (1 = pas du tout, 5 = tout à fait).

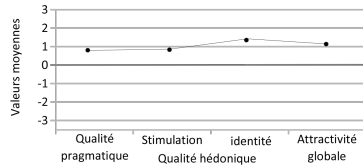


Figure 5. Résultat moyen à l'AttrakDiff [12].

L'ensemble des résultats indique que l'application est satisfaisante pour les vigies, sans toutefois atteindre des scores globaux très élevés. Des améliorations sont donc possibles pour augmenter la motivation, l'UX, et l'utilisabilité dans l'application. Par ailleurs, la Figure 6 illustre l'état de l'engagement des vigies, c'est-à-dire leur fréquence mensuelle d'utilisation de l'application depuis le début de sa mise en place. Nous constatons une forte utilisation en novembre 2013, puis une diminution progressive jusqu'au mois d'août 2014.

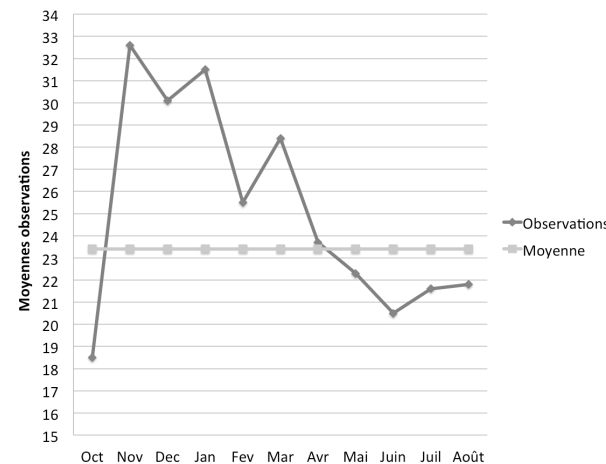


Figure 6. Fréquence moyenne mensuelle d'utilisation de l'application.

Conclusion

Les résultats initiaux de l'étude illustrent la motivation intrinsèque des vigies et le niveau correct d'utilisabilité et d'UX fourni par l'application. Cependant avec la diminution d'utilisation de l'application, il est nécessaire de rapidement analyser les données pour concevoir un nouveau prototype de l'application et tenter de

maintenir l'engagement des vigies avant leur désengagement. Au terme du prototypage, la passation des questionnaires devait être répliquée pour évaluer les améliorations apportées à l'application en termes d'engagement. A travers ce projet, nous espérons compléter et illustrer les liens théoriques qui existent entre les concepts de motivation, d'utilisabilité, d'UX comme éléments de l'engagement.

Cette notion d'engagement revêt une grande importance au sein des interactions hommes-machines. Elle l'est d'une part pour le designer qui peut créer des interfaces plus performantes et engageantes, mais elle l'est aussi pour la création d'interfaces s'adaptant aux caractéristiques des utilisateurs. Des recherches complémentaires sur l'engagement et ses notions connexes permettraient de mieux délimiter ses contours dans le contexte des IHM. L'étude présentée ici se veut un tremplin pour de nouvelles recherches plus approfondies sur cette thématique.

Remerciements

Nous remercions les citoyens-vigies du projet OMNISCIENTIS qui ont accepté de participer à cette étude.

Références

- [1] Y. H. Kim, D. J. Kim, and K. Wachter, "A study of mobile user engagement (MoEN): Engagement motivations, perceived value, satisfaction, and continued engagement intention," *Decis. Support Syst.*, vol. 56, pp. 361–370, Dec. 2013.
- [2] H. L. O'Brien and E. G. Toms, "What is user engagement? A conceptual framework for defining user engagement with technology.," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 6, pp. 938–955, 2008.
- [3] J.-L. Beauvois and R.-V. Joule, *Petit Traité de Manipulation à l'Usage des Honnêtes Gens*, Presses un. 1987.
- [4] H. L. O'Brien and T. E. G, "Examining the generalizability of the User Engagement Scale (UES) in exploratory search," *Inf. Process. Manag.*, vol. 49, no. 5, pp. 1092–1107, 2013.
- [5] R. Ryan and E. Deci, "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions.," *Contemp. Educ. Psychol.*, vol. 25, no. 1, pp. 54–67, Jan. 2000.
- [6] J. Nielsen, *Usability Engineering*, vol. 44. 1993, p. 362.
- [7] V. Roto, "User experience from product creation perspective," *Toward a UX Manif. Work.*, pp. 31–34, 2007.
- [8] M. Hassenzahl, E. L.-C. Law, and E. T. Hvannberg, "User Experience: Towards a unified view," in *UX WS NordiCHI 6*, 2006, vol. 6, pp. 1–3.
- [9] J. B. Pine and J. H. Gilmore, "The experience economy," *Harv. Bus. Rev.*, vol. 7, no. 6, 1998.
- [10] S. Logghe, B. Baccarne, and D. Schuurman, "An exploration of user motivations for participation in Living Labs," in *International Society for Professional Innovation Management Conference*, 2014.
- [11] T. Yang, J. Linder, and D. Bolchini, "DEEP: Design-Oriented Evaluation of Perceived Usability," *Int. J. Hum. Comput. Interact.*, vol. 28, no. 5, pp. 308–346, May 2012.
- [12] M. Hassenzahl, M. Burmester, and F. Koller, "AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität 1 Grundannahmen und Vorarbeiten," pp. 187–196, 2003.
- [13] B. Kim, M. Choi, and I. Han, "User behaviors toward mobile data services: The role of perceived fee and prior experience," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8528–8536, May 2009.

Supervision : contrôler spatialement les objets connectés dans une maison intelligente

Sarthak Ghosh

Télécom-ParisTech, CNRS LTCI
75013, Paris, France
sarthakg91@gmail.com

Eric Lecolinet

Télécom-ParisTech, CNRS LTCI
75013, Paris, France
elc@enst.fr

Gilles Bailly

Télécom-ParisTech, CNRS LTCI
75013, Paris, France
Gilles.bailly@telecom-
paristech.fr

Rémi Sharrock

Télécom-ParisTech, CNRS LTCI
75013, Paris, France
remi.sharrock@telecom-
paristech.fr

Robin Despouys

Télécom-ParisTech, CNRS LTCI
75013, Paris, France
robin.despouys@telecom-
paristech.fr

Résumé

Dans ce papier, nous proposons SuperVision, une nouvelle technique d'interaction pour contrôler à distance les "objets connectés" d'une maison intelligente. Les utilisateurs pointent en direction d'un objet à l'aide d'une télécommande pour visualiser son état et sélectionner ses fonctionnalités. Pour ce faire : 1) nous présentons une nouvelle télécommande qui est augmentée d'un pico-projecteur et d'une glissière ; 2) nous introduisons une visualisation permettant à l'utilisateur de "voir" à travers les murs afin de pouvoir contrôler de la même manière les objets visibles et ceux qui ne sont pas dans son champ de vision ; 3) nous décrivons des applications reposant sur cette technique d'interaction.

Mots clés choisis par les auteurs

Interface Homme Machine ; Maison intelligente ;
Télécommande augmentée ; Projection

Mot clés de la classification ACM

H.5.m. Information interfaces and presentation (e.g.,
HCI): Miscellaneous.

Introduction

Une maison intelligente dispose de nombreux objets connectés tels que des capteurs (de présence, de température...) et des objets contrôlables à distance

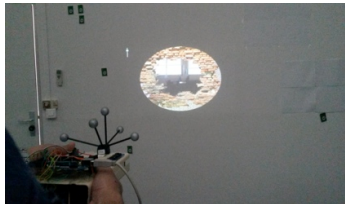


Figure 1. Le concept de SuperVision. A) L'utilisateur pointe en direction d'un objet connecté pour afficher ses caractéristiques et les contrôler. B) L'utilisateur peut zoomer pour voir à travers le mur. C) L'utilisateur peut également voir les objets non visibles à travers un "trou virtuel" dans le mur et les contrôler.

assurant diverses fonctions (éclairage, multimédia, chauffage et climatisation, stores, volets, sécurité...). Ces nouvelles possibilités permettent non seulement d'accroître les services à la personne ou le confort à domicile mais aussi de mieux gérer la consommation d'énergie ou d'optimiser des ressources partagées [11]. Cependant, permettre aux utilisateurs d'accéder et de contrôler efficacement ces dispositifs reste un problème encore largement ouvert.

Une approche relativement classique pour sélectionner et contrôler les objets connectés consiste à utiliser une application sur un dispositif mobile tel qu'un smartphone ou une tablette (e.g. [7][8]). Un modèle de visualisation adapté au dispositif doit alors être construit et la représentation spatiale peut s'avérer être en décalage par rapport à la réalité ou par rapport à la représentation que s'en fait l'utilisateur.

Notre technique, baptisée *SuperVision* s'appuie sur la capacité naturelle des utilisateurs à pointer les objets d'intérêt dans l'espace [6] ainsi que sur leur mémoire visuo-spatiale. L'utilisateur doit simplement pointer en direction des objets connectés à l'aide d'une télécommande augmentée pour les contrôler et accéder aux services associés. Cette télécommande dispose d'un pico-projecteur permettant d'afficher sur l'objet (ou autour de celui-ci) son état courant ainsi que les possibilités de contrôle.

De plus, un avantage de *SuperVision* est de permettre à l'utilisateur de "voir à travers les murs". Cette fonctionnalité inhabituelle vise à lui permettre de contrôler de la même manière les objets visibles et ceux qui ne sont pas dans son champ de vision (typiquement par ce qu'ils sont derrière un mur).

L'utilisateur utilise ainsi la même technique d'interaction s'appuyant sur la mémoire spatiale pour contrôler les objets *dans* et *en dehors* de la pièce où il se trouve. Pour ce faire, nous avons développé une visualisation suggérant à l'utilisateur qu'il voie à travers un "trou virtuel" dans le mur. La taille du trou virtuel s'accroît progressivement comme la "super vision" de Superman.

Nous décrivons dans cet article une implémentation de *SuperVision* où la télécommande dispose d'un pico-projecteur servant à augmenter les objets, d'une glissière pour contrôler la "super vision" ainsi que d'un bouton permettant l'interaction dans les airs. Dans l'implémentation actuelle, la position et l'orientation de la télécommande sont capturées à l'aide d'un dispositif de capture de mouvement ART [2].

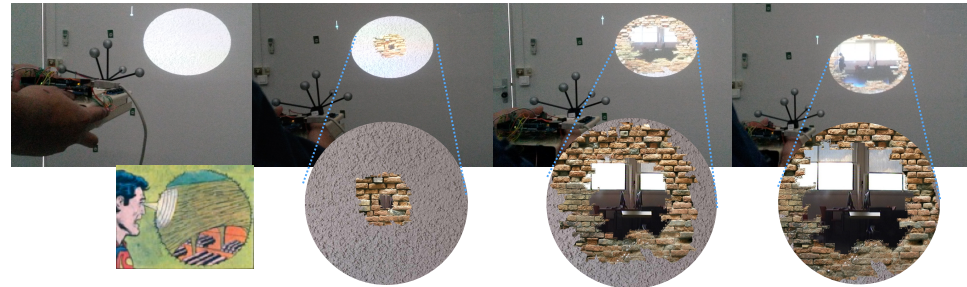
SUPERVISION

SuperVision (Figure 1) est une télécommande pour contrôler des objets connectés dans une maison intelligente. L'utilisateur pointe directement en direction d'un objet pour le contrôler. La télécommande joue le rôle de "torche" virtuelle en ouvrant une fenêtre sur un monde virtuel à l'aide de la vidéo-projection. La télécommande permet alors d'afficher des informations virtuelles sur un objet réel ou dans son voisinage. Le contrôle de cet objet se fait à l'aide d'un bouton situé sur la télécommande ou à l'aide de gestes spatiaux

Visualisation

Augmentation des objets. Pointer un objet permet d'afficher différents types d'informations le concernant, par exemple pour : trouver une commande comme le bouton "Stop" d'un four en marche ; afficher l'état courant de l'objet (consommation énergétique d'un

Figure 2. Les différentes étapes de la « supervision »



radiateur, titre joué sur une chaîne hifi...) ; ou encore afficher des informations plus générales sur la maison (température dans la pièce quand on pointe un radiateur).

Objets non connectés. Un avantage remarquable de la technique proposée est qu'elle peut s'appliquer également pour les objets non connectés. Par exemple, une cheminée ou un poêle classiques peuvent être augmentés en sortie grâce à la vidéo-projection afin d'afficher la température de la pièce lorsque l'utilisateur pointe en direction de ces objets.

Augmentation du voisinage de l'objet. La forme ou la couleur d'un objet peut rendre difficile la projection sur celui-ci. Cependant, les objets sont généralement situés à proximité d'une surface adéquate pour la vidéo-projection (typiquement un mur ou un plafond).

Zoom sémantique. La glissière située sur la télécommande sert à contrôler une fonction de "zoom sémantique" de la quantité d'information affichée. Ainsi, plus l'utilisateur zoome, plus il peut obtenir d'informations sur l'objet pointé. Il peut ainsi obtenir la température courante d'un radiateur, puis, en zoomant, la consommation d'énergie, l'historique de

consommation, les coûts associés ou encore une comparaison avec les statistiques de l'immeuble pour l'inciter à réduire sa consommation énergétique.

Handicap. L'utilisation d'un smartphone ou d'une tablette peut être difficile pour des personnes souffrant de certaines déficiences visuelles. SuperVision permet d'éviter ce problème en affichant si nécessaire les informations souhaitées avec une taille bien plus large.

Interaction avec des objets dans le champ de vision

SuperVision supporte deux types d'interaction pour les utilisateurs novices : le pointage et l'interaction gestuelle.

Pointage. Par défaut, la télécommande n'affiche pas d'information. Lorsque l'utilisateur touche le bouton de la télécommande, celle-ci augmente l'objet pointé en affichant les commandes qui s'y rapportent. Lorsque plusieurs commandes sont affichées dans le faisceau de la télécommande, celle qui est la plus proche du centre du faisceau est en surbrillance. Le bouton de la télécommande permet alors d'activer cette commande. L'utilisateur peut également zoomer à l'aide de la glissière pour permettre une meilleure précision.



Figure 3. Architecture logicielle



pico-projecteur

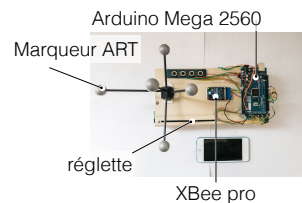


Figure 4. Le prototype de télécommande

Interaction gestuelle. L'utilisateur peut également exécuter une commande à l'aide d'un geste. Plus précisément, une commande est sélectionnée en traversant sa représentation affichée sur le mur avec le bouton de la télécommande. La commande est activée quand le bouton est relâché. L'interaction par franchissement a pour avantage de réduire le niveau de précision requis pour activer les commandes [1].

Interaction experte. Si les commandes sont affichées de manière circulaire, l'utilisateur peut les activer sans utiliser la vidéo-projection. Ce mécanisme est similaire à celui des Marking Menus [4]. En mode novice (mode par défaut), l'affichage des contrôles n'a lieu qu'après que l'utilisateur touche le bouton pendant un court délai (300ms). Si l'utilisateur appuie directement sur le bouton et effectue un geste rapide en maintenant le bouton appuyé, le système passe alors en mode expert. Dans ce cas le menu n'est plus affiché et la commande est exécutée lorsque le bouton est relâché afin de confirmer la sélection.

SuperVision permet aussi à l'utilisateur d'interagir avec des objets situés dans d'autres pièces. Ces pièces nécessitent la présence d'une caméra dont l'image est projetée à l'aide la télécommande sur un mur qui entoure l'utilisateur. La caméra est orientable pour se rapprocher le plus possible de la perspective de l'utilisateur.

Transition fluide. Pour permettre à l'utilisateur d'interagir avec des objets dans une autre pièce, nous avons proposé la métaphore de la "super vision" de superman (Figure 2). La glissière de la télécommande contrôle le zoom sémantique de sorte que l'utilisateur commence à "voir à travers" le mur lorsque l'on

dépasse un certain niveau de zoom, cet effet étant lui-même progressif. Une animation contrôlée par la position de la glissière incite l'utilisateur à continuer d'interagir pour qu'il découvre comment exploiter cette métaphore de "super vision". Il voit d'abord les briques supposées composer l'intérieur du mur, puis un "trou virtuel" apparaît petit à petit et laisse progressivement apparaître la scène derrière le mur. La figure 2 illustre les différentes étapes de cette animation.

Interaction avec des objets derrière le mur. L'utilisateur conserve la même interaction pour contrôler les objets dans la pièce ou dans une autre pièce. L'utilisateur peut donc toujours pointer ou faire des gestes.

Profondeur multiple. La même métaphore peut être utilisée pour voir à travers plusieurs murs. Plus l'utilisateur pousse la glissière, plus il traverse de pièces. Une image peut remplacer le flux vidéo de la caméra pour les pièces sans intérêt ou sans objet connecté (par exemple un débarras).

REALISATION

L'architecture de notre application est illustrée Figure 3. Nous détaillons les éléments de notre prototype.

Télécommande. Notre télécommande (24x12cm) est constituée d'un pico-projecteur Philips PicoPix. En entrée, nous utilisons la technologie ART [2] pour capturer la position et l'orientation de la télécommande et ainsi connaître à tout instant la localisation de la zone projetée dans l'espace physique. Cette technologie nécessite d'instrumentaliser la pièce de plusieurs caméras infrarouge mais a l'avantage d'offrir une bonne précision.

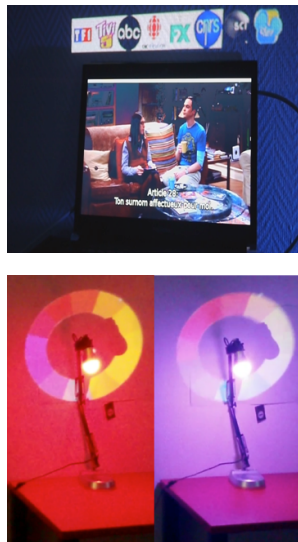


Figure 5. Utilisation de SuperVision avec une télévision interactive (haut) et des lampes multicolores contrôlables (bas)

Nous avons choisi de minimiser le nombre d'interacteurs sur la télécommande afin de permettre à l'utilisateur de l'utiliser sans la regarder [2][4]. Nous avons choisi une réglette pour contrôler le zoom et la profondeur de vue et un seul bouton pour activer les commandes (figure 4). Ce bouton est capacitif afin de différencier l'état touché et appuyé. La vidéo-projection est effectuée uniquement lorsque le bouton est touché afin d'économiser l'énergie. Nous utilisons Arduino pour récupérer l'état du bouton et de la glissière.

Caméra. Nous avons utilisé une caméra Sony qui peut être contrôlée sur deux orientations.

Communication. Le pico-projecteur a son propre système de communication WiFi. Nous utilisons le protocole Xbee pour transmettre les valeurs du bouton et de la glissière. Un protocole simple permet également de connecter des lampes limitlessLED Smart-Light [5] qui sont contrôlables à distance et dont on peut faire varier la couleur et l'intensité.

Logiciel. L'interface est réalisée en C++ avec Qt. Nous avons utilisé OpenCV pour corriger la distorsion des images lorsqu'elles sont projetées sur le mur afin d'obtenir une bonne expérience utilisateur.

Distorsions. Notre système présente plusieurs types de distorsions : 1) les distorsions optiques du projecteur et de la caméra dues à leurs objectifs respectifs ; 2) les distorsions de projection verticale et horizontale, qui entraînent une déformation trapézoïdale (aussi appelée "keystone effect" ou "tombstone effect") résultant de l'écart d'orientation du projecteur par rapport à la normale de la surface de projection ; 3) les distorsions de la captation de la scène par la caméra ; et enfin 4)

la distorsion due à l'écart entre la position des yeux de l'utilisateur et la position du projecteur.

Dans notre système de test, les distorsions de type 1) sont considérées comme fixes et négligeables pour les optiques utilisées. Les distorsions 2) varient fortement en fonction de la position du projecteur. Elles sont corrigées par captation du mouvement et compensées par un algorithme [9][10] écrit en OpenCV. Les distorsions 3) sont compensées par un mouvement en rotation horizontale, rotation verticale et zoom de la caméra. Enfin, la distorsion 4) est considérée comme négligeable.

APPLICATIONS

Nous décrivons ici deux applications

Télévision interactive

La complexité des menus de configuration des télévisions ainsi que le nombre de chaînes TV ne cesse d'augmenter. Pour cette raison, nous utilisons l'espace situé autour de la télévision : en pointant en direction de celle-ci, l'utilisateur découvre les différents choix possibles comme illustré Figure 5-A. Inspiré par [3], cette technique a l'avantage de permettre à l'utilisateur de naviguer dans les différents programmes sans altérer celui qui est en cours.

Ambiance colorée

SuperVision affiche un cercle chromatique autour de chaque lampe. Un geste traversant ce cercle permet de choisir l'ambiance désirée (Figure 5-B).

CONCLUSION ET TRAVAIL FUTUR

Nous avons présenté une nouvelle technique d'interaction pour augmenter et contrôler les objets connectés dans une maison intelligente. Notre approche a consisté à élaborer une télécommande permettant à l'utilisateur de pointer directement l'objet d'intérêt suivant la métaphore de la torche virtuelle. L'originalité de notre technique est de pouvoir contrôler à la fois des objets situés dans le champ de vision de l'utilisateur mais aussi dans d'autres pièces, derrière un mur. Pour ce faire nous avons proposé une métaphore de "super vision" suggérant la possibilité de voir à travers les murs. Un prototype a été développé en ce sens.

Références

- [1] Andrade, J., Meudell, P. Is spatial information encoded automatically in memory? *The Quarterly Journal of Experimental Psychology* 46, 2 (1993), 365-375.
- [2] ART. <http://www.ar-tracking.com/>
- [3] Jones, B., Benko, H. Ofek, E. and Wilson. A. IllumiRoom: peripheral projected illusions for interactive experiences. In *ACM CHI '13*. 2013. 869-878.
- [4] Kurtenbach G. and Buxton, W. Issues in combining marking and direct manipulation techniques. In *ACM UIST'91*. 137-144.
- [5] LimitlessLED Smart Light. <http://www.limitlessled.com>
- [6] Liszkowski, U., Brown, P., Callaghan, T., Takada, A. and vos, C. A prelinguistic gestural universal of human communication. *Cognitive Science*, 39, 678-713.
- [7] Seifried, T., Haller, M., Scott, S., Perteneder, F., Rendl, C., Sakamoto, D., and Inami, M. CRISTAL: a

Nous envisageons comme travail futur de miniaturiser de la télécommande afin de pouvoir entreprendre des tests d'utilisabilité ainsi que de réaliser des évaluations contrôlées pour mieux comprendre l'impact des distorsions sur l'expérience utilisateur et sur la mémoire spatiale. Nous souhaitons également étudier l'utilisation d'images préenregistrées ou le masquage de régions pour pallier aux problèmes de privautés de la vidéo.

Un autre aspect qui nous intéresse concerne l'utilisation de techniques de localisation moins coûteuses et moins invasives que celle que nous utilisons pour notre prototype expérimental afin d'accroître l'acceptabilité de notre approche pour une utilisation réelle.

collaborative home media and device controller based on a multi-touch display. In *ACM ITS'09*, 33-40.

[8] Shoemaker, G., Findlater, L., Dawson, J. Q., and Booth, K. S. 2009. Mid-air text input techniques for very large wall displays. *ACM GI'09*, p. 231-238.

[9] Willis, K., Poupyrev, I. Hudson, S. and Mahler, M. SideBySide: ad-hoc multi-user interaction with handheld projectors. In *ACM UIST'11*, 431-440.

[10] Willis, K. Shiratori, T., and Mahler, M. HideOut: mobile projector interaction with tangible objects and surfaces. In *ACM TEI'13*, 331-338.

[11] Reinisch, Christian, Mario J. Kofler, Félix Iglesias, and Wolfgang Kastner. 2011. "ThinkHome Energy Efficiency in Future Smart Homes." *EURASIP J. Embedded Syst.* 2011 (January): 1:1-1:18.

Une Lampe Anthropomorphique pour la Communication des Emotions

Leonardo Angelini

University of Applied Sciences
and Arts Western Switzerland
Fribourg, Switzerland
leonardo.angelini@hes-so.ch

Maurizio Caon

University of Applied Sciences
and Arts Western Switzerland
Fribourg, Switzerland
maurizio.caon@hes-so.ch

Denis Lalanne

University of Fribourg
Fribourg, Switzerland
denis.lalanne@unifr.ch

Omar Abou Khaled

University of Applied Sciences
and Arts Western Switzerland
Fribourg, Switzerland
omar.aboukhaled@hes-so.ch

Elena Mugellini

University of Applied Sciences
and Arts Western Switzerland
Fribourg, Switzerland
elena.mugellini@hes-so.ch

Résumé

Cet article présente la conception d'une lampe qui permet d'afficher et récolter des états émotionnels de l'utilisateur au travers d'interactions multimodales, basées sur des gestes tangibles du côté utilisateur, et des couleurs et des expressions faciales du côté lampe. En particulier, la lampe bénéficie d'une forme et d'un comportement anthropomorphiques pour rendre l'interaction plus naturelle et spontanée. Deux scénarios d'utilisation sont présentés, ainsi que l'implémentation de l'un d'eux.

Mots clés choisis par les auteurs

Communication médiée par ordinateur ; émotions ; lampe anthropomorphique ; gestes tangibles.

Mot clés de la classification ACM

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

Dans une époque où une bonne partie des communications entre humains est médiée par ordinateur, la communication des états émotionnels est parfois très difficile. La cause est souvent à rechercher dans les moyens de communication qui manquent généralement de toutes les formes de paralangage typiques de la communication entre humains [6]. Par

© ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 26^{ième} conférence francophone sur l'Interaction Homme-Machine, 2014.
<http://dx.doi.org/10.1145/2670444.2670472>



Figure 1. La lampe anthropomorphique dans son état de joie.

exemple, dans les logiciels de messagerie instantanée et dans les réseaux sociaux, les expressions faciales sont limitées à des émoticônes et l'intonation de la voix est souvent mal substituée par la ponctuation ou les émoticônes. Le toucher, stimulé dans la communication face à face avec les gestes de contact, est complètement absent dans les moyens de communication virtuelle les plus utilisés.

Avec le concept d'une lampe anthropomorphique, nous proposons un moyen d'interaction naturel pour communiquer des émotions. Grâce à son comportement et à sa forme anthropomorphique la lampe permet de faciliter la découverte des interactions possibles et la compréhension des états émotionnels affichés. Dans cet article nous montrons le positionnement de la lampe par rapport à l'état de l'art existant et les différentes phases de conception et d'implémentation de la lampe. De plus nous présentons deux scénarios différents d'utilisation : comme compagnon intelligent et comme système de communication à distance de nos émotions. La lampe est montrée sur la Figure 1.

État de l'art

Depuis "Feather, Scent and Shaker" [11], beaucoup de chercheurs ont exploré la possibilité de communiquer des émotions à distance à travers l'utilisation d'objets tangibles. Dans ce contexte, le cadre photo LumiTouch [5] est le premier exemple d'un objet qui permet à la fois une interaction active avec des gestes tactiles sur la surface du cadre photo et une interaction passive grâce à un retour visuel dans l'environnement, qui permet à l'utilisateur d'exploiter la vision périphérique pour connaître l'état d'une personne distante. La lampe anthropomorphique peut être utilisée dans un scénario

similaire mais offre à l'utilisateur une interaction plus naturelle et riche. Le bénéfice d'une interface plus riche pour communiquer des émotions à distance a aussi été démontré dans le projet *cubble* [7]. Dans ce projet, Kowalski et al. ont comparé une interface tangible sous forme de cube avec une interface mobile sur smartphone et ont démontré que l'interface tangible améliorerait l'expérience d'utilisation et emmenait à un échange de messages plus fréquent. Néanmoins, le cube de Kowalski et al. n'offre pas d'affordances anthropomorphiques à l'utilisateur. De plus, les couleurs seules ne permettent pas une codification universelle du message émotionnel envoyé et la plus grande partie des utilisateurs de *cubble* a préféré définir son propre mappage entre couleurs et émotions.

Schmitz [10] a récemment théorisé le bénéfice apporté par les objets qui présentent des similarités avec des êtres vivants dans l'interaction homme-machine. En particulier, les objets qui exposent non seulement une forme mais aussi un comportement zoomorphique ou anthropomorphique arrivent à garantir une interaction plus longue dans le temps et plus intuitive. Schmitz distingue aussi une catégorie d'objets qui sont capables de créer une liaison avec l'utilisateur, grâce au comportement naturel et « life-like » de l'objet. Dans la littérature on peut facilement trouver des compagnons zoomorphiques pour la communication des émotions, par exemple le lapin Nabaztag [8] ou le Hapticat [12]. La lampe ADA, par contre, exploite l'anthropomorphisme pour offrir des affordances à l'utilisateur. En fait, l'utilisateur pourra facilement interagir avec la lampe avec des gestes qu'il ou elle utilise dans sa vie quotidienne, tels que une caresse, un bisou ou une claque. De plus, l'anthropomorphisme de



Figure 2. La lampe anthropomorphique avec ses 5 expressions : contente, triste, confiante (clin d'œil) (2x), extasiée (2x) et fâchée.

la lampe ADA aide à associer l'état affiché sur la lampe à l'état émotionnel d'une personne distante.

Conception d'une lampe anthropomorphique

La conception de la lampe anthropomorphique est le fruit de plusieurs expériences préalables dans la communication des émotions médiée par ordinateur. Suite à divers projets, nous avons mis en place une plateforme multimodale et contextuelle pour la diffusion des états émotionnels dans un environnement intelligent [3]. Cette plateforme bénéficie de divers dispositifs en entrée et en sortie pour partager les émotions dans un environnement intelligent. Ce système exploite le réseau social Twitter comme passerelle pour récupérer et diffuser des messages. Une démonstration de cette plateforme effectuée lors de la conférence ACII 2013 [4], nous a, en particulier, emmené à concevoir la lampe anthropomorphique. Suite à l'analyse de cette démonstration [2], nous avons remarqué que, pour ce qui concerne les dispositifs en sortie, les utilisateurs ont été particulièrement attirés par Aphrodite, un tableau robotique qui reproduit des expressions sur le visage de la Vénus de Botticelli. Une lampe RGB placée à côté, qui changeait de couleur selon les émotions associées aux messages reçus, a été ignorée par la plus grande partie des visiteurs. De plus, pour ce qui concerne les dispositifs en entrée, certains utilisateurs n'ont pas trop apprécié la possibilité de partager les émotions à travers des expressions faciales volontaires (fonctionnalité implémentée dans le smartphone ainsi que dans une télé multitouch).

Suite à cette expérience et en se basant sur l'état de l'art, nous avons donc imaginé une lampe RGB qui puisse représenter les émotions, non seulement à l'aide

des couleurs, souvent difficiles à décoder seules, mais aussi avec des expressions faciales. Dans la Figure 2 nous montrons les différentes expressions faciales codifiées sur la lampe anthropomorphique et les couleurs respectives. Le choix des couleurs est basé sur la roue des émotions de Plutchick [9] comme dans notre précédente lampe RGB [4] (voir aussi Figure 4 a) et b)).

En ce qui concerne le partage d'émotions de la part de l'utilisateur, pour faire face au manque de spontanéité des expressions faciales volontaires, nous avons choisi d'utiliser une interface gestuelle. La spontanéité, dans ce cas, est assurée par l'affordance anthropomorphique de la lampe et par la naturalité des gestes quotidiens que la lampe peut reconnaître. Les gestes que nous souhaitons reconnaître sont caresser, enlacer et embrasser, pour les gestes à valence positive, et donner une claque et secouer pour les gestes à valence négative.

Scénario d'usage

La lampe peut être utilisée dans plusieurs contextes. Dans cet article nous présentons deux applications typiques : le compagnon intelligent et la communication à distance avec une personne chère.

Dans la première application, la lampe anthropomorphique s'anime de sa propre vie et intelligence, réagit aux gestes de l'utilisateur et attire son attention par un comportement qui peut être parfois inattendu, afin de créer un effet de surprise et rendre l'interaction la plus longue possible [10]. Une version avancée (et empathique) de la lampe compagnon pourrait aussi s'adapter à l'état émotionnel de la personne, par exemple avec une analyse

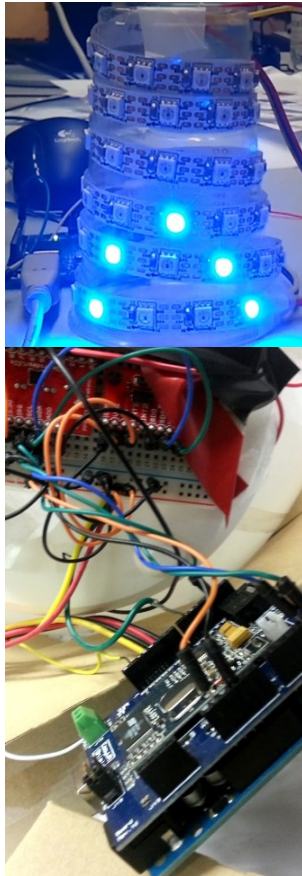


Figure 3. En haut, la structure conique de la bande LED ; en bas, un détail de la carte Arduino+EasyVR et les capteurs MPR121 et ADXL345.

sémantique des messages partagés sur Twitter par son utilisateur. La lampe compagnon pourrait être utilisée tout simplement pour s’amuser, ou pour partager des émotions sur des réseaux sociaux, sans destinataire particulier pour les messages.

Dans la deuxième application, c’est-à-dire la communication à distance des états émotionnels, la lampe agit comme le miroir d’une personne éloignée. La lampe affiche donc l’état émotionnel de cette personne, par exemple récupéré depuis son dernier message sur Twitter ou directement depuis sa lampe. L’état émotionnel de la lampe miroir peut aussi varier selon les gestes que la personne éloignée réalise sur sa lampe : un geste avec valence positive engendrera une émotion de valence positive et vice-versa. De cette façon les deux personnes peuvent communiquer directement, sans devoir écrire leurs états d’âme sur Twitter.

Prototype et fonctionnement dans le scénario d’objet compagnon

La phase de prototypage a été focalisée dans un premier temps sur l’obtention d’une forme anthropomorphique pour la lampe. En gardant la forme ronde et lisse d’une lampe en forme de boule, nous avons modifié son intérieur pour obtenir la forme d’un visage dès que la lampe est allumée. Nous avons substitué l’ampoule originale par une bande de 60 LEDs RGB d’une longueur de 1 mètre. La bande LED a été enveloppée autour d’une structure conique et des compartiments en papier ont été créés pour isoler quatre différentes régions : les deux yeux, une bouche souriante et une bouche froncée. La couleur et l’intensité des LEDs sont contrôlés individuellement par une carte Arduino Uno (Figure 3). De cette façon nous pouvons contrôler la couleur du visage, mais aussi

simuler un clin d’œil et afficher une bouche souriante ou froncée selon la valence de l’expression faciale désirée.

Pour la reconnaissance des gestes, nous avons utilisé le capteur capacitif Freescale MPR121. Nous avons défini 6 zones sur la surface de la lampe et nous l’avons connectée aux entrées correspondantes du capteur MPR121 à travers des électrodes transparentes en PET revêtues d’ITO (Indium Tin Oxide). Les gestes sont reconnus grâce à un simple algorithme à seuil qui mesure le touché sur les différentes zones. Les seuils sont obtenus à travers une calibration automatique du capteur MPR121. Un accéléromètre ADXL345 est utilisé pour détecter des mouvements, en particulier les gestes tels que donner une claque ou secouer.

Le comportement de la lampe a été défini avec un diagramme d’état. Selon l’état actuel, un geste peut emmener la lampe sur un état émotionnel différent, un pour chacune des cinq expressions faciales que la lampe peut afficher. Une absence de gestes, défini dans le système comme inactivité, est aussi lié à un passage d’état. La Figure 4a illustre le diagramme d’état pour les gestes à valence négative et pour l’inactivité. La Figure 4b illustre le diagramme d’état pour les gestes à valence positive. Les flèches multiples de la même couleur impliquent que le geste doit être exécuté plusieurs fois pour permettre le passage d’état (e.g., trois caresses pour passer de la joie à la confiance). Le diagramme d’état peut évidemment être changé pour simuler des comportements différents et prendre en compte aussi des inputs externes comme des messages partagés sur Twitter. Le comportement de la lampe ainsi que la reconnaissance des gestes sont effectués par un logiciel développé en C# et opérant sur un PC

Windows. La carte Arduino est connectée au PC par un câble USB. Une alimentation externe de 5V est nécessaire pour l'alimentation des LEDs RGB.

Travaux futurs et expériences de test

Pour ce qui concerne le développement de la lampe nous avons prévu l'ajout d'un retour haptique (moteur de vibration) et d'un retour sonore, qui sera implémenté avec le module EasyVR pour Arduino. Ce dernier nous permettra aussi de pouvoir reconnaître des « commandes » vocales, pour rendre l'interaction encore plus riche et naturelle. Nous prévoyons de conduire deux types d'expérience : une expérience large public, idéalement lors d'une démonstration dans le cadre de la conférence IHM, où nous mesurerons le temps que chaque utilisateur passera à interagir avec la lampe, le nombre d'interactions différentes découvertes

et le nombre total d'interactions pour chaque type de geste. Nous comparerons aussi les réactions des utilisateurs en utilisant deux diagrammes d'état différents : un premier diagramme d'état où la lampe réagira avec d'expressions faciales à valence positive à des gestes à valence positive (caresse etc.) et un autre où, inversement, la lampe réagira positivement à des gestes à valence négative (claque, etc.). Cette dernière étude est inspirée par l'article provocateur de Buttrick et al. [1]. Le deuxième type d'expérience vise à tester l'utilisation de la lampe compagnon sur une plus longue durée. Nous positionnerons donc la lampe dans un bureau ou dans une salle de séjour et nous étudierons les interactions effectuées pendant une semaine. Cela nous permettra d'établir la longévité de l'attractivité de la lampe et sa capacité d'être un objet de compagnie. Nous visons aussi à implémenter une deuxième lampe

Légende des couleurs des flèches - gestes correspondants.

Figure 4a :

- Noir : Secouer ou donner une claquer
- Bleu : inactivité

Figure 4b :

- Jaune : caresser
- Orange : enlacer
- Rouge : embrasser

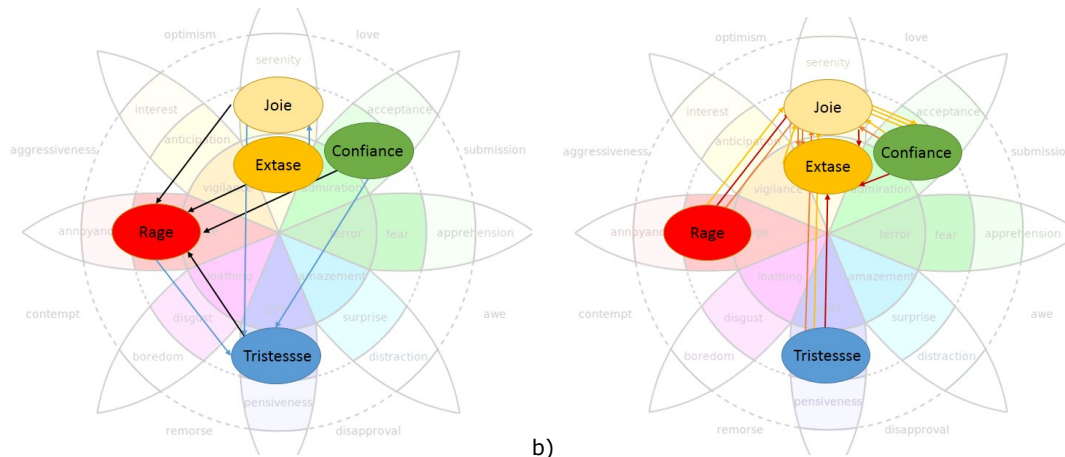


Figure 4. À gauche le diagramme d'état pour les gestes à valence négative, à droite le diagramme pour les gestes à valence positive.

pour tester le scénario de communication à distance et une version de lampe Bluetooth contrôlée depuis le smartphone.

Conclusion

Dans cet article nous avons présenté le concept de la lampe anthropomorphique, une lampe RGB en forme de boule capable de reproduire des expressions faciales et de reconnaître cinq gestes différents: caresser, enlacer, embrasser, donner une claque et secouer. La lampe vise à fournir une interaction plus naturelle et empathique, qui peut être exploitée dans deux scénarios différents: la communication à distance des états émotionnels et l'objet compagnon intelligent. Nous avons implémenté ce dernier scénario et nous

Remerciements

Ce travail a été supporté par la fondation Hasler dans le cadre du projet "Living in Smart Environments".

Références

- [1] Buttrick, L., Linehan, C., Kirman, B., and O'Hara, D. Fifty shades of CHI: the perverse and humiliating human-computer relationship. In *CHI EA'14* (pp. 825-834). 2014. ACM.
- [2] Caon, M., Angelini, L., Khaled, O. A., Lalanne, D., Yue, Y., and Mugellini, E. Affective Interaction in Smart Environments. *Procedia Computer Science*, 32, pp. 1016-1021. 2014.
- [3] Caon, M., Angelini, L., Yue, Y., Khaled, O. A., and Mugellini, E. (2013). Context-Aware Multimodal Sharing of Emotions. In *Proc. Of HCII'13* pp. 19-28. 2013. Springer Berlin Heidelberg.
- [4] Caon, M., Khaled, O. A., Mugellini, E., Lalanne, D., and Angelini, L. Ubiquitous Interaction for Computer Mediated Communication of Emotions. In *Proc. of ACII'13* (pp. 717-718).

montrons le fonctionnement de la lampe dans ce cadre. Des améliorations de la lampe sont évidemment encore envisagées : un nombre important d'électrodes transparentes devrait être intégré sur la lampe, si possible avec des techniques de revêtement en ITO directement sur la vitre de la lampe ; la forme des parties du visage devrait être améliorée, ainsi que l'uniformité des couleurs et l'homogénéité de l'éclairage ; pour finir, la prise en compte des informations de contexte devrait permettre d'améliorer l'empathie générée par la lampe. Des expériences d'utilisation ont été prévues pour la suite du projet, en particulier lors d'une démonstration dans le cadre de la conférence IHM.

- [5] Chang, A., Resner, B., Koerner, B., Wang, X., and Ishii, H. LumiTouch: an emotional communication device. In *CHI'01 EA* (pp. 313-314). 2001. ACM
- [6] Clubb, O. L. Human-to-Computer-to-Human Interactions (HCHI) of the communications revolution. *Interactions*, 14(2), 35-39. (2007).
- [7] Kowalski, R., Loehmann, S., and Hausen, D. cubble: a multi-device hybrid approach supporting communication in long-distance relationships. In *Proc. of TEI'13*. pp. 201-204. ACM.
- [8] Peters, L. Nabaztag wireless communicator. *Personal Computer World*, 2. 2006.
- [9] Plutchik, R. The Nature of Emotions, *American Scientist*, 89(4), 344-350, 2001.
- [10] Schmitz, M. Concepts for life-like interactive objects. In *Proc. of TEI* (pp. 157-164). ACM.
- [11] Strong, R., and Gaver, B. Feather, scent and shaker: supporting simple intimacy. In *Proc. of CSCW'96*. 1996 Vol. 96, No. 96, pp. 29-30.
- [12] Yohanan, S., Chan, M., Hopkins, J., Sun, H., and Maclean, K. Hapticat : Exploration of Affective Touch. *Proc. ICMI'05*, ACM (2005), 222-229.

Index des auteurs

- A** Mourad Abed, 183
Omar Abou Khaled, 207
Jonathan Aceituno, 55
Carlos Agon, 1
Rossana Maria de Castro Andrade, 183
Leonardo Angelini, 207
- B** Gilles Bailly, 9, 38, 105, 201
Eric Barboni, 80
Sandra Bardot, 165
Marie-Hélène Bekaert, 189
Yacine Bellik, 130
Carla Ilane Moreira Bezerra, 183
Anke Brock, 165
- C** François Cabestaing, 171, 189
Gaëlle Calvary, 123, 153
Maurizio Caon, 207
Sheelagh Carpendale, 113
Géry Casiez, 28, 147
Stéphane Chatty, 60
Kolski Christophe, 177
Stéphane Conversy, 60, 70
Etienne Corteel, 105
Martin Cronel, 80
- D** Yannick Deleris, 80
Mustapha Derras, 91
Robin Despouys, 201
Pierre Dragicevic, 48
Emmanuel Dubois, 91
David Duprat, 18
Alban Duprès, 171
- E** James Eagan, 9
Houcine Ezzedine, 183
- F** Jean-charles Fabre, 80
Camille Fayollas, 80
Humbert Fiorino, 123
Raphaël Foulon, 105
- G** Yoann Gabillon, 123
Hind Gacem, 9
Jérémie Garcia, 1
Hélène Gaspard-Boulinç, 60, 70
Yoann Gauthier, 18
Lilian Genaro Motti, 139
Jérémie Gilliot, 147
Asma Gharsellaoui, 130
Alix Goguey, 28
Philippe Gorce, 139
Sarthak Gosh, 201
Thomas J. Grabowski, 113
Guillaume Gronier, 195
- H** Alexis Heloir, 101
Nathalie Henry Riche, 113
Christophe Hurter, 18
- J** Christophe Jacquet, 130
Thibaut Jacob, 105
Francis Jambon, 153
Christophe Jouffrais, 165
- K** Christophe Kolski, 101
- L** Denis Lalanne, 207
Eric Lecolinet, 9, 105, 201
Rémi Lesbordes, 18
Catherine Letondal, 18

Index des auteurs

- M** Wendy Mackay, 1
Tara Madhyastha, 113
Sylvain Malacria, 38
Káthia Marçal de Oliveira, 177
Joran Marcy, 18
José Maria Monteiro, 183
Elena Mugellini, 207
- N** David Navarre, 80
Van Bao Nguyen, 153
Fabrizio Nunnari, 101
- P** Philippe Palanque, 80
Alexis Paoleschi, 18
Gary Perelman, 91
Charles Perin, 48
Yvan Peter, 159
Charlotte Piau, 189
Célia Picard, 91
Thomas Pietrzak, 28, 38
- Patricia Plénacoste, 159
- R** Mathieu Raynal, 91
Yann Riche, 113
José Rouillard, 171
Nicolas Roussel, 28, 55, 113, 147
- S** Ismayle Santos, 183
Rainara Maia Santos, 183
Ahmed Seffah, 177
Marcos Serrano, 91, 165
Rémi Sharrock, 201
Lepreux Sophie, 177
- T** Isabelle Tissier, 195
Theophanis Tsandilas, 1
- V** Nadine Vigouroux, 139
Jean-luc Vinot, 18, 60
Daniel Vogel, 28

