

EICS 2020

12th ACM SIG CHI Symposium on

Engineering Interactive Computing Systems

Proceedings

Judy Bowen, Jean Vanderdonckt, Marco Winckler
(editors)



SIGCHI

Special Interest Group for
Computer-Human Interaction



Association for
Computing Machinery

Sophia Antipolis, France ◆ June 23-26, 2020

The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York New York 10121-0701

ACM COPYRIGHT NOTICE. Copyright © 2020 by the Association for Computing Machinery, Inc.
© 2020 Copyright is held by the owner/author(s).

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 third-party components of this work must be honored. For all other uses, contact the Owner/Author.

ACM ISBN **978-1-4503-7984-7/20/06**.

TABLE OF CONTENTS

Foreword

Late-Breaking Results

The Foldinterface Editor: A Visual Tool for Designing User Interfaces for Foldable Displays

Jean Vanderdonckt, Iyad Khaddam, and Radu-Daniel Vatavu

Mind the (Persuasion) Gap: Contrasting Predictions of Intelligent DSS with User Beliefs to Improve Interpretability

Michael Chromik, Florian Fincke, and Andreas Butz

Model Voyager: Visualization of CRF UI Models

Iyad Khaddam and Jean Vanderdonckt

Computing Aesthetics of Concrete User Interfaces

Patrick Mbenza and Nicolas Burny

Recognizing a Sketched Task Model by Multi-Stroke Gesture Recognition

Jorge-Luis Perez-Medina and Karina-Beatriz Jimenes-Vargas

A Pen User Interface for Controlling a Virtual Puppet

Jean Vanderdonckt and Radu-Daniel Vatavu

SeqCheck: A Model Checking Tool for Interactive Systems

Jessica Turner, Judy Bowen, and Steve Reeves

How Many Steps to Represent Individual Gait?

Lilian Genaro Motti Ader, Killian McManus, Barry R. Greene, and Brian Caulfield

Development Framework for Context-Aware Augmented Reality Applications

Sarah Claudia Krings, Enes Yigitbas, Ivan Jovanovikj, Stefan Sauer, and Gregor Engels

Meta-Level Support for Facilitating Participation in Website (Re-)Design Activities

Henrik Hertel, Anke Dittmar, and Doritt Linke

Closed Loop Feedback Nudges Using Nerve Stimulation

Rohan Hundia, Aditya Vijayvargiya, and Aaron Quigley

Doctoral Consortium

Engineering Privacy-aware Smart Home Environments

Akshika Wijesundara

Towards Supporting Reproducibility of Experimental Studies in GUI Visual Design

Nicolas Burny

Method for Developing Virtual Reality Applications for Cognitive Intensive Training Tasks

Ketoma Vix Kemanji

Crowd-powered Creativity Support Systems

Jonas Oppenlaender

Tutorial

Task Models Based Engineering of Interactive Systems

Celia Martinie and Philippe Palanque

Workshop

Entrain - Exploring New Territorial User Interfaces

Anne-Marie Dery-Pinna, Philippe Renevier, Marco Winckler, Christophe Kolski, Sophie Lepreux, and Jean Vanderdonckt

EICS 2020 Committee

FOREWORD

The ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS) is an international conference devoted to engineering usable and effective interactive computing systems. EICS focuses on models, languages, notations, methods, techniques, and tools for supporting the development life cycle of interactive systems at any stage, from requirements elicitation, specification, design, development, validation, evaluation, and deployment. In more general terms, the scientific and technical contributions that address the development life cycle of interactive systems and associated user interfaces are relevant to EICS. The scientific field and community of EICS are typically located at the overlapping of Human-Computer Interaction (HCI) and Software Engineering (SE).

The EICS conference started as a sponsored ACM SIGCHI conference in 2009, but its journey as a scientific venue is probably one of the longest in the scientific field of Human-Computer Interaction. EICS was created as a continuation and merge of a set of series of conferences, symposiums and workshops, including:

- TAMODIA, the Annual Conference on Task Models and Diagrams, organized annually from 2002 to 2009.
- EHCI, the IFIP International Conference on Engineering for Human-Computer Interaction, which was previously organized in 2001, 2004 and 2007.
- DSV-IS, the Design, Specification, and Verification of Interactive Systems International Conference, organized annually from 1994 to 2008. DSV-IS was mainly sponsored by Eurographics and its proceedings were published in the Lecture Notes in Computer Science series from Springer, most of the time.
- CCL, the IFIP Working Conference on Command Languages, with the first edition organized in 1974.

EICS 2020 was planned as the 12th edition of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, to be held from June 23-26, 2020. EICS 2020 would have hosted by the University Côte d’Azur (Polytech Nice, SophiaTech Campus). Very unfortunately, due the unexpected consequences of the Covid-19 pandemic, the physical meeting had to be cancelled to ensure the safety of EICS community. Nonetheless, all contributions reviewed and accepted for EICS 2020 are published as usual. Authors of full papers, late-breaking results and tutorials, who would have presented their work during 2020 conference have instead been invited to present their work at EICS 2021, which is going to take place in Eindhoven, Netherland, in June 2021. The Doctoral Consortium and one workshop accepted for EICS 2020 occur remotely on June 23 as planned. The other tracks (posters, demos and tech notes) were cancelled prior to the submission deadline.

EICS 2020 received 61 full paper submissions during three reviewing rounds (Q3 and Q4 2019 and Q1 2020), of which 18 were accepted representing an acceptance rate of 30%, which is similar to last year’s edition. Whilst full papers accepted at EICS 2020 are published in the PACMHCI journal, this volume features the contributions received in the form of 11 Late-Breaking Results (from 18 submissions, representing 61% acceptance rate), 4 Doctoral Consortium (4 contributions), the summary of one Tutorial called “Task Models Based Engineering of Interactive Systems”, and of the workshop entitled “Entrain, Exploring New Territorial User Interfaces”.

We would like to express our gratitude to those who supported EICS 2020 in many different ways starting with the chairs and technical program chairs of EICS 2021 for proposing a venue for presenting full papers, late-breaking results and tutorial that could not be presented this year. Many thanks to the authors for authors, members of the program committee and reviewers, who continued working despite the uncertainty associated with the occurrence of conference. A special thought of gratitude for the chairs of tracks Full Papers (Jean Vanderdonckt and Judy Bowen), Late-Breaking Results (Célia Martinie and Panos Markopoulos), Workshops (Damiano Distanto and Steeve Reeves), Tutorials (Anke Dittmar and Hermann Kaindl), Doctoral Consortium (Sophie Dupuy-Chessa and Jürgen Ziegler), whose contributions are available here. Also, many thanks to Luciana Zaina, our publicity chair, who help to disseminate the call for papers that bring you all here. Last, but not least, we would like to thank the executive committee of EICS for their support in helping to explore all the alternatives available, including the negotiations with the EICS 2021 chairs.

Judy Bowen, Jean Vanderdonckt, and Marco Winckler

EICS 2020 Committee

General Conference Chair

Marco Winckler, *Université Côte d'Azur, France*

Full Paper Chairs

Judy Bowen, *University of Waikato, New Zealand*

Jean Vanderdonckt, *Université catholique de Louvain, Belgium*

Late-Breaking Results Chairs

Panos Markopoulos, *Eindhoven University of Technology, Netherlands*

Célia Martinie, *Université Paul Sabatier – Toulouse III, France*

Tech Notes Chairs

Vivian Genaro Motti, *George Mason University, United States of America*

Demos/Poster Chairs

Félix Albertos Marco, *Universidad de Lleida, Spain*

Jan Van Den Bergh, *Expertise Centre for Digital Media, UHasselt, Belgium*

Technical and accessibility Chairs

Lilian Genaro Motti, *University College Dublin, Ireland*

Publicity Chair

Luciana Zaina, *University Federal University of São Carlos, Brazil*

Workshop Chairs

Damiano Distanto, *University of Rome Unitelma Sapienza, Italy*

Steeve Reeves, *University of Waikato, New Zealand*

Tutorial Chairs

Anke Dittmar, *University of Rostock, Germany*

Hermann Kaindl, *TU Wien, Vienna, Austria*

Doctoral Consortium Chairs

Sophie Dupuy-Chessa, *Grenoble Institute of Technology & LIG, France*

Jürgen Ziegler, *University of Duisburg-Essen, Germany*

Local organizers

Olivier Corby, *INRIA Sophia Antipolis, France*

Corinne Jullien-Haddad, *Université Côte d'Azur, France*

Philippe Renevier Gonin, *Université Côte d'Azur, France*

EICS Steering Committee chair

José Creissac Campos, *Universidade do Minho, Portugal*

Late-Breaking Results Committee

Matthias Book, *University of Iceland, Iceland*

Michael Burmester, *IXD - Stuttgart Media University, Germany*

Diogo Cabral, *ITI/LARSyS, IST, University of Lisbon, Portugal*

António Coelho, *University of Porto, Portugal*

José Creissac Campos, *Universidade do Minho, Portugal*

Olfa Daassi, *École Nationale d'Ingénieurs de Carthage, Tunisia*

Anke Dittmar, *University of Rostock, Germany*

Sophie Dupuy-Chessa, *Grenoble Institute of Technology & LIG, France*

Peter Forbrig, *University of Rostock, Germany*

Mathias Funk, *Eindhoven University of Technology, Netherlands*

Vivian Genaro Motti, *George Mason University, United States of America*

Sebastian Günther, *Telecooperation Lab, TU Darmstadt, Germany*

Nicolas Hili, *IRT Saint Exupery, France*

Jun Hu, *Eindhoven University of Technology, Netherlands*

Shah Rukh Humayoun, *San Francisco State University, United States of America*

Rajiv Khadka, *Idaho National Laboratory, United States of America*

Vassilis-Javed Khan, *Eindhoven University of Technology, Netherlands*

Suzanne Kieffer, *Université catholique de Louvain, Belgium*

Min Hun Lee, *Carnegie Mellon University, United States of America*

Min Li, *TU/e, Netherlands*

Ivan Logre, *Laboratoire Informatique de Grenoble (LIG), France*

Nathan Magrofuoco, *Université catholique de Louvain, Belgium*

Joseph Maguire, *University of Glasgow, United Kingdom*

Marco Manca, *HIIS Laboratory, CNR-ISTI, Italy*

Panos Markopoulos, *Eindhoven University of Technology, Netherlands*

Célia Martinie, *Université Paul Sabatier – Toulouse III, France*

Jorge-Luis Perez-Medina, *Universidad de Las Américas, Ecuador*

Matti Pouke, *University of Oulu, Finland*

Teresa Romão, *Universidade Nova de Lisboa, Portugal*

Carmen Santoro, *ISTI-CNR, Pisa, Italy*

Fereshteh Shahmiri, *Georgia Institute of Technology, United States*

Jose Luis Silva, *Instituto Universitário de Lisboa, Portugal*

Lucio Davide Spano, *University of Cagliari, Italy*

Konstantinos Tsiakas, *Eindhoven University of Technology, Netherlands*

Jan Van den Bergh, *Expertise Centre for Digital Media, UHasselt, Belgium*
Jean Vanderdonckt, *Université catholique de Louvain, Belgium*
Benjamin Weyers, *Trier University, Germany*
Marco Winckler, *Université Cote d’Azur, France*
Enes Yigitbas, *Paderborn University, Germany*
Bin Yu, *Eindhoven University of Technology, Netherlands*
Jürgen Ziegler, *University of Duisburg-Essen, Germany*

Full papers Committee Members (Associated Chairs)

Simone Barbosa, *PUC-Rio, Brazil*
Arnaud Blouin, *Université Rennes, IRISA, France*
Judy Bowen, *University of Waikato, New Zealand*
Gilbert Cockton, *University of Sunderland, United Kingdom*
Emmanuel Dubois, *University of Toulouse, IRIT - Elipse, France*
Sophie Dupuy-Chessa, *Grenoble Institute of Technology & LIG, France*
Anke Dittmar, *University of Rostock, Germany*
Peter Forbrig, *University of Rostock, Germany*
Elizabeth Furtado, *Universidade de Fortaleza, Brazil*
Lilian Genaro Motti, *University College Dublin, Ireland*
Christian Kray, *University of Münster, Germany*
Kris Luyten, *Expertise Centre for Digital Media, UHasselt, Belgium*
Célia Martinie, *Université Paul Sabatier – Toulouse III, France*
Panos Markopoulos, *Eindhoven University of Technology, Netherlands*
Jeff Nichols, *Google, United States of America*
Laurence Nigay, *LIG - Université Grenoble Alpes, France*
Philippe Palanque, *ICS, University Toulouse III, Toulouse, France*
Fabio Paternò, *ISTI-CNR, Pisa, Italy*
Steve Reeves, *University of Waikato, New Zealand*
Carmen Santoro, *ISTI-CNR, Pisa, Italy*
Lucio Davide Spano, *University of Cagliari, Italy*
Jan Van den Bergh, *Expertise Centre for Digital Media, UHasselt, Belgium*
Jean Vanderdonckt, *Université catholique de Louvain, Belgium*
Radu-Daniel Vatavu, *University Stefan cel Mare of Suceava, Romania*
Marco Winckler, *Université Cote d’Azur, France*
Luciana Zaina, *University Federal University of São Carlos, Brazil*
Jürgen Ziegler, *University of Duisburg-Essen, Germany*

Full papers Reviewers

Julio Abascal

Pierre Akiki

Baris Aksanli

Eric Barboni

Benoît Bossavit

Jacky Bourgeois

Elodie Bouzekri

Karin Coninx

Luigi De Russis

Tim Donkers

Bruno Dumas

Florian Echter

Josefina Guerrero Garcia

Isabela Gasparini

Michael Harrison

Ebba Hvannberg

Christopher Johnson

Jessica Korte

Rossitza Kotelova

Katharina Krösl

Jair Leite

Luis Leiva

Sophie Lepreux

Zhen Li

Dawei Liang

Nathan Magrofuoco

Christian Meurisch

Giulio Mori

Elena Navarro

Thanh-Diane Nguyen

Jonas Oppenlaender

Ignacio Panach

Oscar Pastor

Anne-Marie Pinna-Dery

Susanne Putze

Gustavo Rossi

Anne Roudaut

Guilherme Schardong

Vinícius Segura

Beat Signer

Simone Stumpf

Kashyap Todi

Santiago Villarreal

Xuhai Xu

Enes Yigitbas

Alexandre Demeure

Thanh-Diane Nguyen

Jorge Perez-Medina

Angel Puerta

José Creissac Campos

Helen Petrie

Vangel Ajanovski

Wei Bai

Karima Boudaoud

Giorgio Brajinik

Anders Bruun

Rute C. Sofia

Francisco Maria Calisto

Stéphane Chatty

Olfa Daassi

Nitish Devadiga

Hans Gellersen

Rachid Gherbi

Yolanda Gil

Juan Manuel Gonzalez Calleros
Pascual Gonzalez
Miriam Greis
Taisa Guidini Gonçalves
Jan Gulliksen
Xin Guo
Jürgen Hahn
Syed Shabih Hasan
Florian Heller
Kasper Hornbæk
Chyng-Yang Jang
Hermann Kaindl
Suzanne Kieffer
Mohamed Lachgar
Gierad Laput
Jonathan Lazar
Min Hun Lee
Ye Liu
Victor Lopez
María Dolores Lozano
Jamie Mahoney
V. K. Chaithanya Manam
Sven Mayer
Gerrit Meixner
Christos Mousas
Jaime Munoz
David Navarre

Ngoc Nguyen
Kathia Oliveira
Dan Olsen
Alex Olwal
Mehdi Ousmer
Nadia Pantidi
Vipul Raheja
Hajo Reijers
Anne Roudaut
Silvia Rueda
Thouraya Sboui
Vinoth Pandian Sermuga Pandian
Fereshteh Shahmiri
Jan Stage
Eleni Stroulia
Krishna Subramanian
Harold Thimbleby
Stefan Trausan-Matu
Umair ul Hassan
Virag Varga
Matthias Weber
Liqi Xu Xu
Wenge Xu
Iman Yeckehzaare
Sang Ho Yoon
Tengxiang Zhang
Kening Zhu

The Foldinterface Editor: A Visual Tool for Designing User Interfaces for Foldable Displays

Jean Vanderdonckt

Université catholique de Louvain
Louvain Research Institute in
Management and Organizations
Louvain-la-Neuve, Belgium
jean.vanderdonckt@uclouvain.be

Iyad Khaddam

Université catholique de Louvain
Louvain Research Institute in
Management and Organizations
Louvain-la-Neuve, Belgium
iyad.khaddam@uclouvain.be

Radu-Daniel Vatavu

MintViz Lab,
MANSiD Research Center
University Ștefan cel Mare of Suceava
Suceava 720229, Romania
radu.vatavu@usm.ro

ABSTRACT

Designing foldable user interfaces (“foldinterfaces”) is complex and time-consuming because of the multiple technologies involved in the process, from the hardware details for folding pixels to design requirements regarding efficiency and ease of use. To assist this process, we introduce a visual editor for designers to specify foldinterfaces by implementing an extension of the Yoshizawa-Randlett diagramming system for origami and Event-Condition-Action rules from event-driven software architecture. The outcome is rendered as a 3-D foldable surface in a virtual environment.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; *Graphical user interfaces*; *Virtual reality*; • **Software and its engineering** → **Integrated and visual development environments**;

KEYWORDS

Foldable displays; Foldable user interfaces; Foldinterfaces; Graphical User Interfaces; Origami; Kirigami; Visual editor; Prototyping; Tool.

ACM Reference Format:

Jean Vanderdonckt, Iyad Khaddam, and Radu-Daniel Vatavu. 2020. The Foldinterface Editor: A Visual Tool for Designing User Interfaces for Foldable Displays. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398490>

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398490>

1 INTRODUCTION

Unlike conventional screens, foldable displays present the attractive feature that they can fold into small form factors when not used at full display capacity and, consequently, they optimize the display area and physical space to accommodate a variety of desktop, mobile, and wearable contexts of use [2, 10, 12, 14, 18, 22, 24]. Computer displays have improved considerably by incorporating new technologies, including liquid crystals, LEDs, organic LEDs, and active-matrix organic LEDs [23]. Moreover, the recent availability of commercial foldable smartphones¹ suggests a realistic future for thinner, lighter, and reconfigurable computer displays that can accommodate a diversity of form factors and contexts of use where screen space is defined *relative* to the task and is no longer absolute.

The theoretical concept of a “foldable display” has an important role in this process by enabling on-the-fly adaptation of the display form factor, such as by means of gesture input [5], and by supporting new interactions besides conventional input techniques [2]. Foldable displays act as both observation and action surfaces according to the classification of Coutaz *et al.* [4] as users interact with the graphical user interface (GUI) by touch and/or gesture input and the effects are rendered on the same surface. Foldable displays offer a large interaction surface when unfolded and reduce to a minimum size when stored away. They also offer new opportunities for rich interactions, such as by means of “folding gestures,” and greater adaptability to the context of use.

However, designing user interfaces for foldable displays is conditioned by the availability of the physical hardware, which hinders prototyping and evaluation. To address this aspect, we introduce the “Foldinterface Editor,” a visual editor that enables specification of the GUI of foldable interactive displays and surfaces and visualizes the result in the form of a virtual surface that unfolds in 3-D. The Foldinterface Editor makes prototyping less expensive and less dependent on the availability of hardware and, thus, fosters new explorations of design possibilities for user foldinterfaces.

¹<https://www.samsung.com/global/galaxy/galaxy-fold>

2 RELATED WORK

Probably the first manifestation of a folding digital surface is identifiable in the “Fold & Drop” interaction technique of Dragicevic [5] from 2004, where flipping between the windows of the GUI was achieved through “folding” and “dropping” gestures instead of pressing keys on the keyboard to imitate real-world flipping of pages from a book. Soon after, two families of foldable GUIs emerged: *projection-based systems*, where the elements of the GUI are projected onto a foldable, uninstrumented surface that is tracked in space, and *foldable devices*, where the surface and display are integrated in the device itself. For example, GUIs can be video projected to various surfaces, such as scrolls, fans, and even umbrellas as demonstrated by Lee *et al.* [14]; PaperWindows [11] introduced digital paper by means of video projections onto a sheet of paper tracked by a motion capture system; and, in FlexPad [22], interactive content was displayed onto a foldable surface tracked by a Microsoft Kinect sensor mounted on the ceiling. MigriXML [17] graphically rendered user interfaces across multiple interactive surfaces in Virtual Reality (VR) and enabled migration with adaptation of UIs from one surface to another. FoldMe [12] projected a GUI on predefined hinges where IR markers served as reference points. Instead of projecting GUIs on foldable surfaces, devices were used to control a virtual scene in which the GUI was rendered: in Gallant *et al.* [6], a paper sheet with IR reflectors was tracked by a video camera and folding gestures were recognized; by means of the augmented sheet of paper, the user could control the GUI displayed on a conventional screen.

Other approaches to foldable displays considered extensions of display devices or their integration into complex form factors. For example, WristOrigami [26] was designed to extend the interaction surface of a smartwatch by connecting foldable displays on its four sides with rotary hinges and rubber strings. Various 2-D and 3-D configurations were enabled to support interactive tasks, such as the “cushion” fold, “cupboard” fold, “book” fold, and the “cube” fold [26]. The SurfaceConstellations technique and system [15] adopted a complementary approach by assembling individual displays into complex form factors, such as a prism.

Yet other approaches have considered tools for prototyping such interfaces. For example, Tangrami [25] provided a digital design toolkit in which the shape and functionality of a foldable UI could be specified with building blocks designed in a visual editor and subsequently printed and assembled. The connection between the foldable UIs and the prototype application was managed via the Open Sound protocol [25]. Foldios [18] are foldable interactive objects with embedded input sensing and output capabilities made out of thin-film printed electronics. Foldable maps [16] imitate real-world physical maps on which buildings are recreated in 3-D.

3 A VISUAL NOTATION SYSTEM FOR PROTOTYPING USER FOLDINTERFACES

In this section, we introduce a visual notation for foldinterfaces inspired by the practice of origami [13]. First, we present our definition for foldinterfaces.

Definition

In this work, we use the term “foldinterfaces” to denote all user interfaces designed for foldable interactive displays and surfaces, no matter the technology employed to display the user interface (*e.g.*, video projection), the capabilities embedded in the surface itself (*e.g.*, thin-film printed electronics), or the structure employed to compose multi-display systems that can change shape (*e.g.*, constellations of displays).

The Yoshizawa-Randlett diagramming system

We rely in this work on the Yoshizawa-Randlett diagramming system [13], a standard notation for making origami that specifies several basic operations; see Figure 1:² *valley fold* (a part of the model is folded onto another to form a “valley” shape; the valley fold is represented by a dashed line with an arrow pointing in the direction of the fold); *mountain fold* (one part of the model is folded onto the back of another to form a “mountain” shape; the mountain fold is represented by a dashed dotted line and with an empty arrow pointing in the direction of the fold); *fold and unfold* (one part of the model is folded and then unfolded using the valley fold; the same operation can be performed with a mountain fold); *turn over* (the invisible lines behind the model are represented with dotted lines); *rotate* (the model is rotated up to a specified angle in the direction indicated by the arrows).

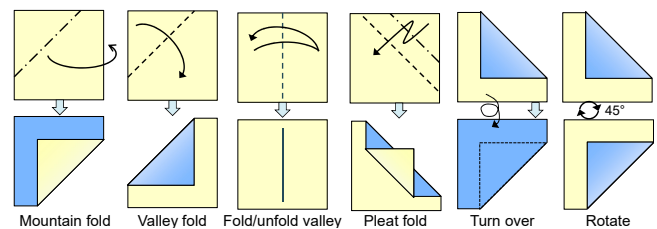


Figure 1: The Yoshizawa-Randlett notation for origami [13].

A Visual Notation for Foldinterfaces

The Yoshizawa-Randlett notation takes into account folding operations on surfaces that follow predetermined lines, as shown in Figure 1. However, foldinterfaces also present bending capabilities. To accommodate the latter, we introduce an extension of the Yoshizawa-Randlett notation for the scope

²Images redrawn from Yrogirg (Sarnitskiy Grigory), released into public domain; https://en.wikipedia.org/wiki/Yoshizawa-Randlett_system

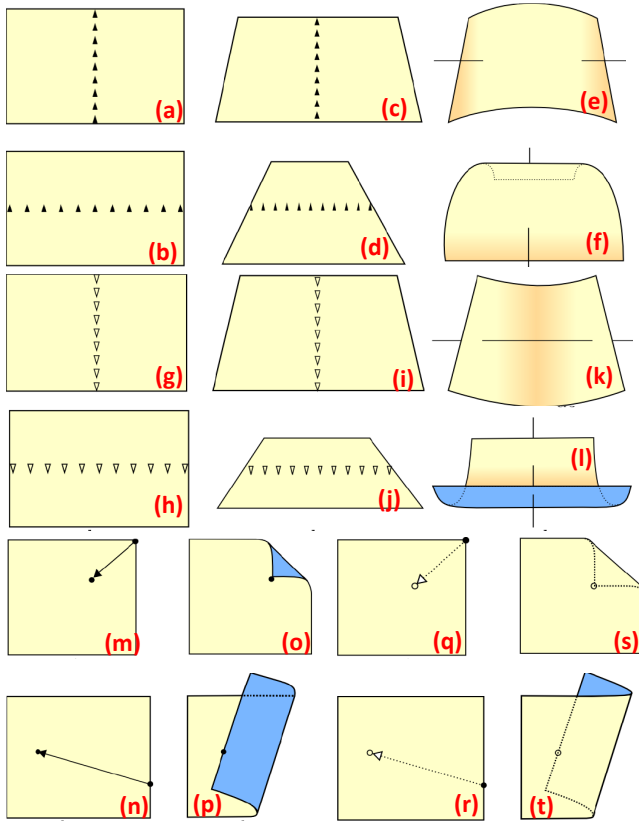


Figure 2: Our visual system for foldinterfaces: *mountain bending* (a and b denote flat states; c and d add perspective; e and f show the result of the bending); *valley bending* (g and h are flat states; i and j show perspective; k and l show the result); *valley corner-bending* (m, n flat state; o, p the result); *mountain corner-bending* (q, r flat state; s and t the result).

and purpose of foldinterfaces (see Figure 2) that consists of the following operations:

- *Mountain bending*: the surface bends upwards to form a “mountain”-like shape. The corresponding visual notation is a series of arrows pointing upwards.
- *Valley bending*: the surface bends downwards.
- *Valley corner-bending*: the valley-bending operation is applied to any part of the surface. The corresponding visual notation is a straight arrow (as opposed to a curved arrow for the Yoshizawa-Randlett valley fold). The starting point is denoted by a small circle that indicates where the surface that must be picked up to perform the bending operation. The ending point is also denoted with a circle and corresponds to the point on the surface where dragging must end.
- *Mountain corner-bending*: the mountain-bending operation is applied to any part of the surface. The corresponding notation is a dotted straight empty arrow.

4 THE FOLDINTERFACE EDITOR

Based on the visual notation system illustrated in Figure 2, we developed the Foldinterface Editor that enables prototyping of user foldinterfaces in four steps:

- (1) *Surface creation*. The surface is specified using direct manipulation, e.g., two rectangular surfaces can be joined to form a newspaper-like shape; see Figure 3a.
- (2) *Definition of content*. Graphical content is imported and graphical widgets (e.g., dials, radio buttons, push buttons, etc.) are positioned on top of the surface; see Figure 3b.
- (3) *Definition of the interaction*. A set of Event-Condition-Action (ECA) rules [27] are specified; see Figure 3c. Each rule is composed of an *event* (which defines the signal that triggers the application of the rule), a *condition* (specifies when the corresponding action is executed), and an *action* (denotes the command to be executed when the condition is evaluated to true).
- (4) *Automated generation of the foldinterface*. Once the previous steps are completed, the Foldinterface Editor automatically generates X3D code³ for a virtual foldable surface corresponding to the specified foldinterface and renders it in a X3D-compliant browser for further examination and evaluation; see Figure 3d.

Conceptual Modeling in the Foldinterface Editor

The conceptual modeling of a foldinterface consists of two meta-models expressed as UML Class Diagrams V2.5 [3] and stored as XML files in the Foldinterface Editor. Transformation templates [1] are used to convert the XML files into X3D code via one-to-one mappings. Figure 4 illustrates the conceptual model of a foldable display (Foldable-Display), which is composed of a set of Surfaces following the definition of interaction surfaces of Coutaz *et al.* [4]. Multiple Foldable-Displays link to each other through Fold-Links, a super-class for all the types of foldable operations illustrated in Figure 2. The Content rendered on a Foldable-Display is subject to the User-Interaction model that implements a set of ECA-Rules. For each rule, Events are decomposed into Event-Types and Event-Sources. A Condition is decomposed into three parts, as follows: the Condition-Object specifying the object for which the condition is evaluated, the Condition-Value specifying the value that is compared against, and the Condition-Operator denoting the operator used to compare the condition object against the expected value. An Action is decomposed into the Action-Object, Action-Type, and the Action-Value. Figure 3 shows the steps for creating a newspaper foldinterface based on the “fold-unfold” gestures from Figure 6. When the rules are specified, the foldinterface is automatically rendered in X3D and becomes actionable by

³<https://www.web3d.org/x3d/what-x3d/>

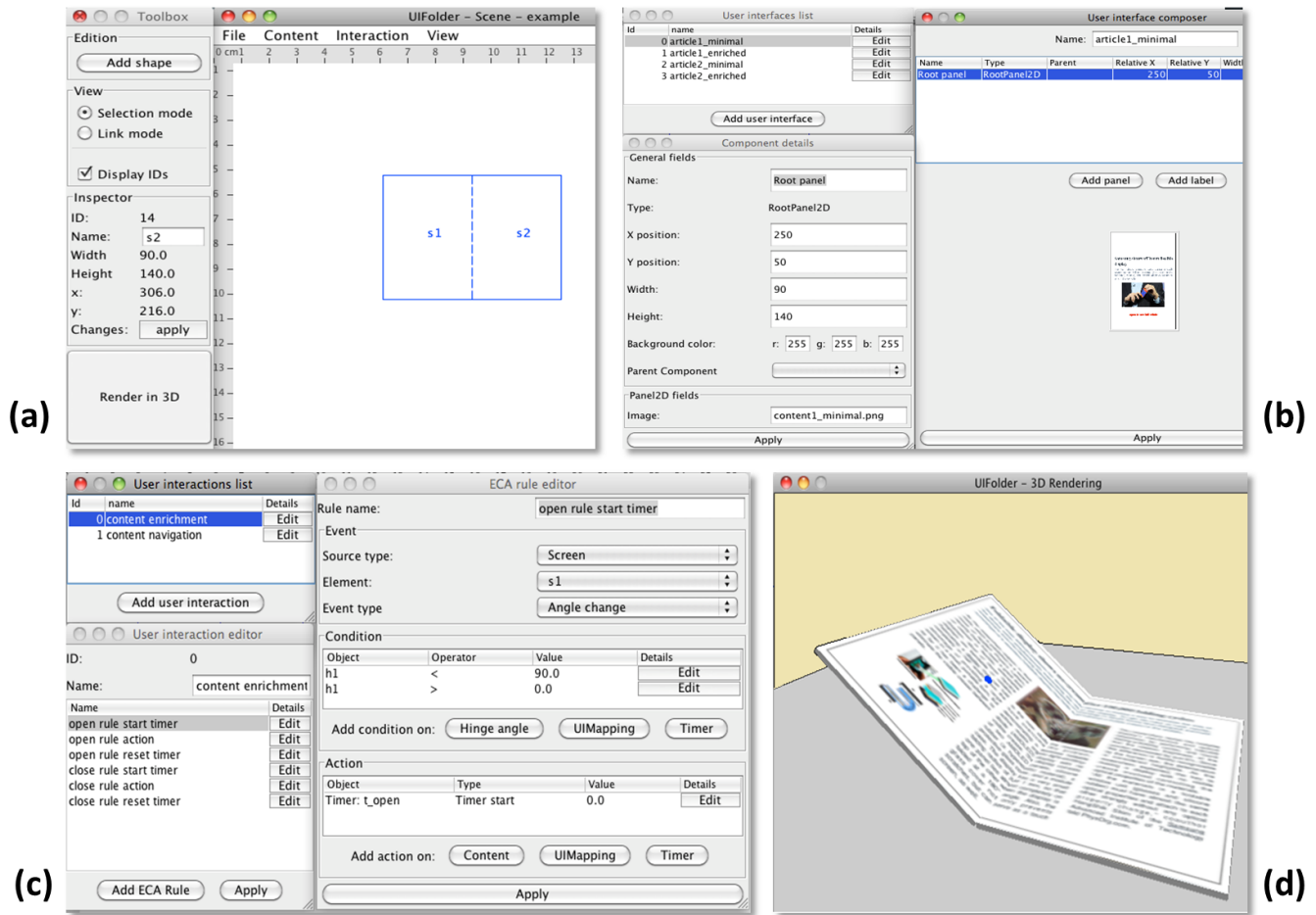


Figure 3: Example of a scenario for a foldable GUI; see also Figure 7.

means of the ECA rules. Figure 7 shows the same fold-unfold behavior, but with standard GUI widgets, such as text fields, spin buttons, and dials.

Implementation

We developed the Foldinterface Editor using Java 7 SDK with a total number of 186 expandable classes, of which 115 classes dedicated to GUI widgets and content for the foldable surfaces. At the moment, only rectangular surfaces can be defined with respect to the fold links and composed by joining their edges by means of direct manipulation. The current version of the Foldinterface is available on GitHub at the web address <https://github.com/jeanvdd/UIFolder>.

5 CONCLUSION AND FUTURE WORK

We introduced the Foldinterface Editor, a visual tool for prototyping designs of foldable graphical user interfaces that renders the results in X3D for visualization in a web browser. There are several directions for future development: building a catalog of foldable patterns that emerge from the practice

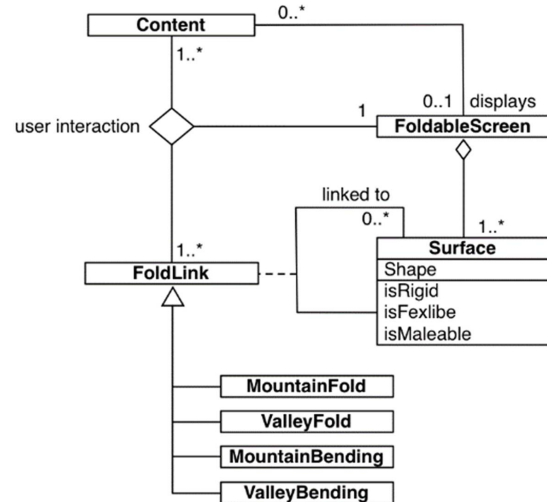


Figure 4: Conceptual model for a foldinterface.

of using foldinterfaces and extending our visual notation system and the rendering engine of the Foldinterface Editor to support bendable and rollable user interfaces. For

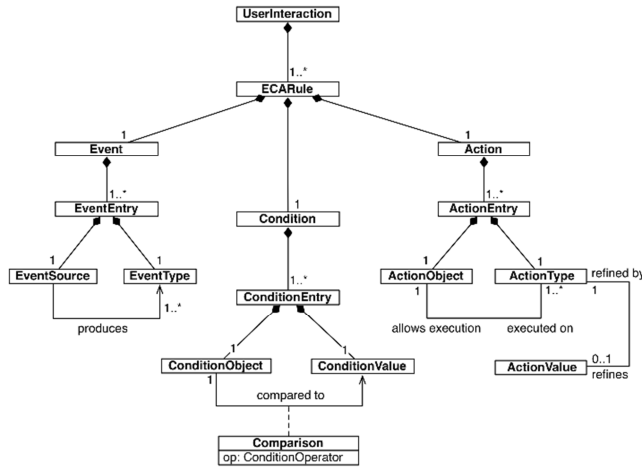


Figure 5: Model for interaction with foldinterfaces.

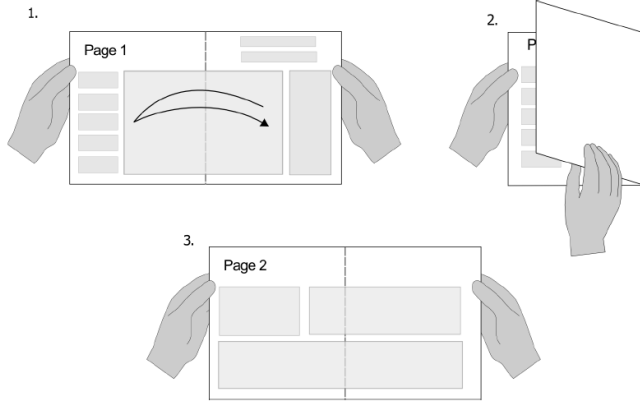


Figure 6: Content navigation on a newspaper-like shape foldinterface using a bimanual fold-unfold gesture.

instance, we would like to relax the current restriction in the Editor regarding rectangular surfaces by considering simple quadrilaterals, both concave (e.g., a dart) and convex (e.g., a trapezoid), but also complex quadrilaterals, such as antiparallelograms. The ultimate goal is to reach the capability to render any GUI on any surface or set of connected surfaces. Exploration of various types of gestures for interacting with foldinterfaces, such as hand and finger gestures sensed by smart rings [7], including bimanual gestures [8], and integrating foldable displays into multi-device architectures for smart environments [19, 20] represent interesting directions for future exploration as well.

Another direction for future work will explore more connections with the practice of Origami and Kirigami, respectively. The original art of Origami forbids cutting, gluing, or marking. When this constraint is relaxed, the Japanese word “Kirigami” is used instead. With the advent of shape-changing UIs, future work will investigate to what extent a

Kirigami-like GUI could be prototyped toward novel foldinterfaces, such as cell-composed touchscreen surfaces [9] and modular devices that afford lending [21]. Also, the origami practice indicates a catalog of basic folds, of which some have direct and meaningful application to foldinterface as we showed in this work, while others are purely decorative. Therefore, we plan to compile an inventory of folds that are meaningful for user foldinterfaces inspired by origami galleries⁴ and catalogs of foldable structures.

ACKNOWLEDGMENTS

The authors would like to thank Noé Phan for developing the Foldinterface Editor.

REFERENCES

- [1] Nathalie Aquino, Jean Vanderdonckt, and Oscar Pastor. 2010. Transformation Templates: Adding Flexibility to Model-Driven Engineering of User Interfaces. In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10)*. ACM, New York, NY, USA, 1195–1202. <https://doi.org/10.1145/1774088.1774340>
- [2] Jesse Burstyn, Amartya Banerjee, and Roel Vertegaal. 2013. FlexView: An Evaluation of Depth Navigation on Deformable Mobile Devices. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. ACM, New York, NY, USA, 193–200. <https://doi.org/10.1145/2460625.2460655>
- [3] Ashley Bush and Sandeep Purao. 2011. Mapping UML Techniques to Design Activities. In *Information Modeling in the New Millennium*, Matti Rossi and Keng Siau (Eds.). IDEA Publishing Group, Chapter 12, 199–217.
- [4] Joëlle Coutaz, Christophe Lachenal, and Sophie Dupuy-Chessa. 2003. Ontology for Multi-surface Interaction. In *Proc. of IFIP TC13 Int. Conf. on Human-Computer Interaction, 1st-5th Sept. 2003, Zurich (INTERACT'03)*. IOS Press, 447–454.
- [5] Pierre Dragicevic. 2004. Combining Crossing-based and Paper-based Interaction Paradigms for Dragging and Dropping Between Overlapping Windows. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 193–196. <https://doi.org/10.1145/1029632.1029667>
- [6] David T. Gallant, Andrew G. Seniuk, and Roel Vertegaal. 2008. Towards More Paper-like Input: Flexible Input Devices for Foldable Interaction Styles. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 283–286. <https://doi.org/10.1145/1449715.1449762>
- [7] Bogdan-Florin Gheran, Jean Vanderdonckt, and Radu-Daniel Vatavu. 2018. Gestures for Smart Rings: Empirical Results, Insights, and Design Implications. In *Proc. of the 2018 Designing Interactive Systems Conference (DIS '18)*. 623–635. <https://doi.org/10.1145/3196709.3196741>
- [8] Bogdan-Florin Gheran, Radu-Daniel Vatavu, and Jean Vanderdonckt. 2018. Ring X2: Designing Gestures for Smart Rings Using Temporal Calculus. In *Proceedings of the 2018 ACM Conference Companion Publication on Designing Interactive Systems (DIS '18 Companion)*. 117–122. <https://doi.org/10.1145/3197391.3205422>
- [9] Alix Goguey, Cameron Steer, Andrés Lucero, Laurence Nigay, Deepak Ranjan Sahoo, Céline Coutrix, Anne Roudaut, Sriram Subramanian, Yutaka Tokuda, Timothy Neate, Jennifer Pearson, Simon Robinson, and Matt Jones. 2019. PickCells: A Physically Reconfigurable Cell-Composed Touchscreen. In *Proceedings of the 2019 CHI*

⁴<https://www.giladorigami.com/origami-galleries.html>

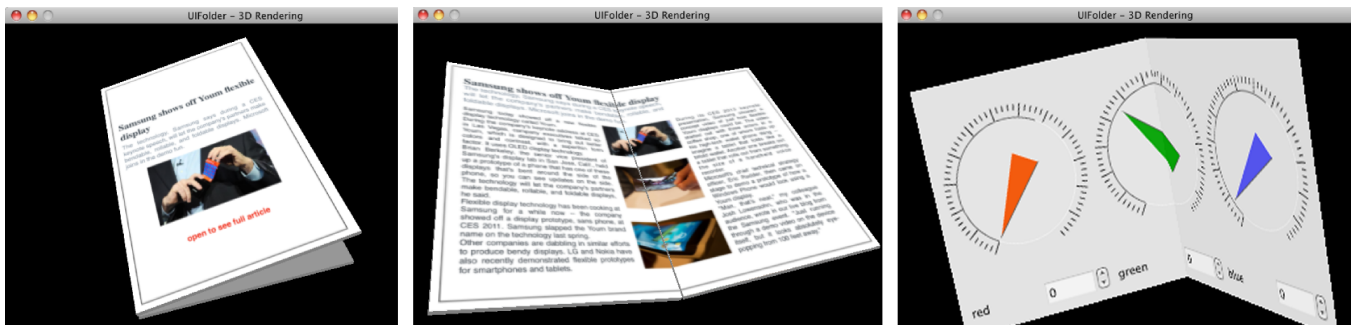


Figure 7: X3D representation of the foldinterface from Figure 6 in our Foldinterface Editor.

- Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 273, 14 pages. <https://doi.org/10.1145/3290605.3300503>
- [10] Kim Gustafson, Oyuna Angatkina, and Aimy Wissa. 2019. Model-based design of a multistable origami-enabled crawling robot. *Smart Materials and Structures* 29, 1 (nov 2019), 015013. <https://doi.org/10.1088/1361-665x/ab52c5>
- [11] David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. 2005. Paper Windows: Interaction Techniques for Digital Paper. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 591–599. <https://doi.org/10.1145/1054972.1055054>
- [12] Mohammadreza Khalilbeigi, Roman Lissermann, Wolfgang Kleine, and Jürgen Steimle. 2012. FoldMe: Interacting with Double-sided Foldable Displays. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, New York, NY, USA, 33–40. <https://doi.org/10.1145/2148131.2148142>
- [13] Robert J. Lang. 2011. Origami Diagramming Conventions - Diagramming, Part 1. Web site. Retrieved November 26, 2018 from <http://langorigami.com/article/origami-diagramming-conventions/>.
- [14] Johnny C. Lee, Scott E. Hudson, and Edward Tse. 2008. Foldable Interactive Displays. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 287–290. <https://doi.org/10.1145/1449715.1449763>
- [15] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. 2018. SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 354, 14 pages. <https://doi.org/10.1145/3173574.3173928>
- [16] S. Martedi, H. Uchiyama, G. Enriquez, H. Saito, T. Miyashita, and T. Hara. 2010. Foldable augmented maps. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. 65–72. <https://doi.org/10.1109/ISMAR.2010.5643552>
- [17] José Pascual Molina Massó, Jean Vanderdonckt, and Pascual González López. 2006. Direct Manipulation of User Interfaces for Migration. In *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI '06)*. ACM, New York, NY, USA, 140–147. <https://doi.org/10.1145/1111449.1111483>
- [18] Simon Olberding, Sergio Soto Ortega, Klaus Hildebrandt, and Jürgen Steimle. 2015. Foldio: Digital Fabrication of Interactive and Shape-Changing Objects With Foldable Printed Electronics. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 223–232. <https://doi.org/10.1145/2807442.2807494>
- [19] Ovidiu-Andrei Schipor, Radu-Daniel Vatavu, and Jean Vanderdonckt. 2019. Euphoria: A Scalable, Event-Driven Architecture for Designing Interactions Across Heterogeneous Devices in Smart Environments. *Information and Software Technology* 109 (2019), 43–59. <https://doi.org/10.1016/j.infsof.2019.01.006>
- [20] Ovidiu-Andrei Schipor, Radu-Daniel Vatavu, and Wenjun Wu. 2019. SAPIENS: Towards Software Architecture to Support Peripheral Interaction in Smart Environments. *Proceedings of the ACM on Human-Computer Interaction* 3, EICS, Article Article 11 (June 2019), 24 pages. <https://doi.org/10.1145/3331153>
- [21] Teddy Seyed, Xing-Dong Yang, and Daniel Vogel. 2017. A Modular Smartphone for Lending. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 205–215. <https://doi.org/10.1145/3126594.3126633>
- [22] Jürgen Steimle, Andreas Jordt, and Pattie Maes. 2013. Flexpad: Highly Flexible Bending Interactions for Projected Handheld Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 237–246. <https://doi.org/10.1145/2470654.2470688>
- [23] Miriam Sturdee and Jason Alexander. 2018. Analysis and Classification of Shape-Changing Interfaces for Design and Application-Based Research. *ACM Comput. Surv.* 51, 1, Article Article 2 (Jan. 2018), 32 pages. <https://doi.org/10.1145/3143559>
- [24] Aneesh P. Tarun, Peng Wang, Audrey Girouard, Paul Strohmeier, Derek Reilly, and Roel Vertegaal. 2013. PaperTab: An Electronic Paper Computer with Multiple Large Flexible Electrophoretic Displays. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 3131–3134. <https://doi.org/10.1145/2468356.2479628>
- [25] Michael Wessely, Nadiya Morenko, Jürgen Steimle, and Michael Schmitz. 2018. Interactive Tangrami: Rapid Prototyping with Modular Paper-folded Electronics. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings (UIST '18 Adjunct)*. ACM, New York, NY, USA, 143–145. <https://doi.org/10.1145/3266037.3271630>
- [26] Kening Zhu, Morten Fjeld, and Ayça Ünlüer. 2018. WristOrigami: Exploring Foldable Design for Multi-Display Smartwatch. In *Proc. of the 2018 Designing Interactive Systems Conference (DIS '18)*. ACM, New York, NY, USA, 1207–1218. <https://doi.org/10.1145/3196709.3196713>
- [27] Michael Zoumboulakis, George Roussos, and Alex Poullovassilis. 2004. Active Rules for Wireless Networks of Sensors & Actuators. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*. ACM, New York, NY, USA, 263–264. <https://doi.org/10.1145/1031495.1031527>

Mind the (Persuasion) Gap: Contrasting Predictions of Intelligent DSS with User Beliefs to Improve Interpretability

Michael Chromik
LMU Munich
Munich, Germany
michael.chromik@ifi.lmu.de

Florian Fincke
LMU Munich
Munich, Germany
florian.fincke@campus.lmu.de

Andreas Butz
LMU Munich
Munich, Germany
butz@ifi.lmu.de

ABSTRACT

Decision support systems (DSS) help users to make more informed and more effective decisions. In recent years, many intelligent DSS (IDSS) in business contexts involve machine learning (ML) methods, which make them inherently hard to explain and comprehend logically. Incomprehensible predictions, however, might violate the users' expectations. While explanations can help with this, prior research also shows that providing explanations in all situations may negatively impact trust and adherence, especially for users experienced in the decision task at hand. We used a human-centered design approach with domain experts to design a DSS for funds management in the construction industry and identified a strong need for control, personal involvement, and adequate data. To create an adequate level of trust and reliance, we contrasted the system's predictions with the values derived from an analytic hierarchical process (AHP), which makes the relative importance of our users' decision-making criteria explicit. We developed a prototype and evaluated its acceptance with 7 construction industry experts. By identifying situations in which the ML prediction and the domain expert potentially disagree, the DSS can identify a persuasion gap and use explanations more selectively. Our evaluation showed promising results and we plan to generalize our approach to a wider range of explainable artificial intelligence (XAI) problems, e.g., to provide explanations with arguments tailored to the user.

CCS CONCEPTS

• **Human-centered computing** → **HCI design and evaluation methods**.

KEYWORDS

decision support systems; decision-making; interpretability; explainable artificial intelligence; analytical hierarchical process

ACM Reference Format:

Michael Chromik, Florian Fincke, and Andreas Butz. 2020. Mind the (Persuasion) Gap: Contrasting Predictions of Intelligent DSS with User Beliefs to Improve Interpretability. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia

Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398491>

1 INTRODUCTION

The rapidly growing volume of data in many parts of the enterprise makes it necessary to structure and manage it in information systems. Those systems which help in decision-making are referred to as *decision support systems (DSS)* [30]. With the recent improvements in *machine learning (ML)* methods, DSS are becoming more and more intelligent. So-called *intelligent decision support system (IDSS)* augment the collected data with predictions that guide and (semi-) automate parts of the decision-making process [30]. However, these intelligent DSS also introduced new challenges because their rationale is often not interpretable and hence perceived as non-deterministic by their users.

The effectiveness of an intelligent DSS depends not only on the accuracy of its underlying ML model or algorithm. Instead, it is only effective if it serves the information needs of decision-makers and is also accepted and trusted by them. Jarvis describes DSS as the general idea of "*combining the computer's computational power with the decision maker's intuition and judgment in an interactive manner, [so that] better decisions will result than by either the computer or human taken separately*" [11]. To achieve such a symbiosis, we need to design user interfaces (UI) that communicate the rationale behind algorithmic predictions in human-understandable terms. The UI should help to calibrate the user's understanding of the system's capabilities and limitations to prevent both over-reliance (when users blindly trust system recommendations) and under-reliance (when users simply ignore system recommendations) [5].

We conducted a design study in the construction industry and asked decision-makers about their requirements regarding interpretability of a novel intelligent DSS module on addenda approval. We use the term *interpretability* to refer to measures provided by a DSS with the aim of enabling users to understand its inner workings. Interpretability is a broad concept that may imply other distinct ideas such as transparency, trust, and fairness [17]. It is often used to indirectly evaluate whether important requirements, such as reliability, trust, or control are met in a particular context [8]. Biran and Cotton consider intelligent systems interpretable "*if their operations can be understood by a human*" [3]. We followed a human-centered design process to understand how project managers and executives make decisions regarding validation and approval of budget addenda. Budget management in the construction industry is an interesting context to study for two reasons: First, the construction industry itself is one of the least digitized industries but digitization efforts (e.g., building information modeling (BIM)) are

Permission to make digital or hard copies of all or part 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 the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7984-7/20/06...\$15.00
<https://doi.org/10.1145/3393672.3398491>

gaining adoption despite decision makers skepticism [1, 4]. Secondly, the addenda approval process is a complex decision situation that requires decision makers to retrieve and interpret data from distributed sources and also consider their dependencies. To date this is a highly manual and subjective process

This paper investigates interpretability needs of human decision-makers in the field regarding an intelligent DSS. In particular, we propose an approach to align the level of trust and reliance by contrasting ML predictions with user beliefs. User beliefs can be extracted through multi-attribute decision making approaches such as the *analytic hierarchical process (AHP)*. Making the user beliefs explicit allows the system to better identify *persuasion gaps* [6], i.e., situations in which the system and user base their decision on different criteria. We think that this approach might be a valuable starting point for providing selective and personalized explanations to the field of explainable artificial intelligence (XAI). With this work, we put our suggested approach and formative evaluation up for discussion with our fellow researchers.

2 RELATED WORK

2.1 Intelligent Decision-Support Systems

Decision-making refers to the cognitive process of selecting a logical choice from many available alternatives. Decision-making problems are often structured into three phases: problem identification, model development and use, and action plan development [21]. In our work, we focus on the second phase that deals with eliciting user preferences and comparing alternatives in a consistent way. If a decision is rational it is typically based on facts instead of arbitrary choices. *Multi-attribute decision making (MADM)* describes approaches that leverage (potentially conflicting) attributes to select, compare, and rank a limited number of discrete alternatives [31]. The rationality of decision-making, however, is bounded as individuals are often not able to make optimal decisions in an economically rational way due to cognitive limitations and resource constraints [28]. Simon suggests that instead of maximizing (search for the best possible option), decision makers in the field are rather satisficing (stick to an option that is considered good enough) [28].

In many business-related contexts, *decision support systems (DSS)* organize relevant facts to assist users in decision-making and improve effectiveness of the decision outcome [30]. DSS can range from simple spreadsheets to complex data warehouse systems [30]. They are typically distinguished by their underlying technology, theory foundations, target users, and decision tasks [2]. So called, *intelligent decision support systems (IDSS)* use artificial intelligence methods to support the decision-making and exhibit some notion of "intelligent behavior" [30]. Such intelligent behavior may either be applied to the system's underlying data base (e.g., identifying relevant attributes), knowledge base (e.g., suggesting decision alternatives), or model base (e.g., choosing applicable formal decision-making methods) [22].

In our work, we focus on IDSS that recommend decision alternatives to the user (*model development and use at the knowledge base*). The first generation of IDSS (also called *knowledge-driven DSS*) leveraged domain knowledge encoded in rule-based reasoning modules [30]. Examples include MYCIN [27] for bacterial infection

diagnosis or DENDRAL [16] for chemical analysis. In contrast, modern IDSS leverage machine learning (ML) methods that implicitly infer rules from observations and thus learn from experience. This implicit inference of rules may result in the *black-box problem* for decision-makers. A black box refers to a situation in which it is possible to observe the input and outputs of a model, but the internals remain disclosed or uninterpretable to humans. In ML, the black box behavior may arise either from complex algorithms (as with deep neural networks) or from proprietary models that may otherwise be interpretable (such as with the COMPAS recidivism model) [24]. As decision-makers were always considered an integral part of the DSS [22], special attention must be paid to the design of the user interaction. With ML-enabled DSS this interaction must include explanation facilities that result in usable interpretability for decision makers.

2.2 Interpretability and Task Expertise

Prior research shows that a lack of interpretability can lead to users that mistrust, misuse, or reject a system [15, 19]. Often these result from a perceived mismatch between users' expectations and the actual behavior of a system [9]. Chander et al. describe two reasons for the mismatch to occur in a business-related decision-making context [6]: (i) the system's underlying data lacks decision criteria relevant for this situation (*awareness gap*). For instance, the user might have relevant contextual information from a phone call with a client that the system has no access to; (ii) the system's prediction is not in line with the user's beliefs and the system fails to persuade the user to adjust their beliefs (*persuasion gap*). In such a situation, the user and the system have access to the same information but weight decision criteria differently. The gaps are even more pronounced in a business-related context, where domain experts often can draw upon rich prior knowledge and beliefs about the decision situation when assessing the system (*extrinsic setting*) [20]. Explanations about the factors that contributed to the system's prediction, e.g., in natural language or in the form of visualizations, are considered one way of addressing those gaps. However, in prior research, rational explanations were shown to be only effective for participants that are not familiar with a given task [26]. The effects of explanation drop as users' confidence with the task increases over time. As user get confident with the task and the system's prediction, they become less situation aware. Most explanation approaches assume that explanations are displayed with every system prediction.

3 USE CASE AND METHODOLOGY

In our work, we outline and probe an approach that provides system explanations only when a mismatch with the user's beliefs occurs (persuasion gap). Such an approach may increase the situation awareness of decision-makers. We focused on the use case of addenda approval and risk assessment in the construction industry. We cooperated with Alasco¹ and their clients. Alasco provides a web-based cost controlling system for the construction industry that connects stakeholders and digitizes processes around budgeting, reporting, addenda management, and payment. We followed a human-centered design process that consisted of three

¹<http://www.alasco.de>

phases: (i) we interviewed executives about their current addenda-related decision-making and derived intelligibility needs for an (semi-)automated addenda approval process; (ii) we designed and developed an interactive prototype that reflects those intelligibility needs; (iii) we evaluated our prototype in a formative user study to understand the acceptance of the prototype workflow.

Use Case: Addenda Approval. Our use case targets project management (PM) executives in the construction industry. The PM is responsible for the fulfillment of the construction project and acts as a coordinator between contractors on behalf of the building owner [13]. During the initial budget planning, the overall budget is split into a hierarchy of cost groups (e.g., property or financing). Each cost group consists of one to many contract units. Each contract unit represents the budget planned for commissioning a contractor for a task. As a construction project advances, contract units might require budget addenda due to unforeseen incidents or flaws in the initial budget planning. After ensuring the plausibility of the addendum, PM executives need to redistribute budgets from other contract units to accommodate the addendum. While doing this, decision-makers need to take the addendum risk and cost forecast of the other contract units into account.

Phase I: Need-Finding. The goal of the first phase, was to identify decision criteria and interpretability needs for an intelligent DSS for the addenda approval process. To understand domain experts' current decision-making processes around addenda approval, we conducted semi-structured interviews with 3 project managers and 2 project controllers who are proficient users of the Alasco software. Their average industry experience were 2.8 years (min=1, max=6). The interviews were held in the regular work environments and took between 30 and 45 minutes. The interviews were recorded and transcribed. To enrich our qualitative insights, participants were surveyed after each interview with the *decision-making questionnaire (DMQ)*. The DMQ is a validated psychological questionnaire that aims to examine factors important to a decision-maker in the moment of decision-making in a specific context [7]. It consists of 14 questions which correspond to 3 factors (and 10 subfactors) that characterize a decision-making situation: (i) the nature of the decision or task, (ii) the cognitive and affective abilities of the decision maker, and (iii) the environment of the decision.

Phase II: Prototyping. We integrated our prototype as a separate module on top of the Alasco software. We reused the general structure and user interface of the software as participants were already familiar with it. The prototyping process was informed by the results of the need analysis as well as prior work on DSS and interpretability. Financial data has strict privacy regulations. Also, the production data of the participant's organizations varied greatly and was often incomplete. Thus, we centered our prototype around an addenda approval scenario based on a synthetically created data set so that all participants could be evaluated on the same scenario. The scenario consisted of an onboarding phase and an addenda approval phase. We developed a functional front-end prototype while the back-end was mostly static around the evaluation scenario.

Phase III: Formative Evaluation. After the design phase, we conducted a formative user study to evaluate the prototype's acceptance regarding participant's sense of control and sense of information.

As the use case and required domain expertise limited the number of potential study participants, we adopted a qualitative evaluation approach. We recruited 5 project managers and 2 project controllers with an average industry experience of 3.7 years (min=0.5, max=15). The user study included 3 participants from phase I as well as 4 new participants. This reduced the risk of receiving biased feedback from participants who had already thought about (semi-)automating addenda approvals. The participants were presented with the scenario and asked to complete an addenda approval task including the onboarding task. While doing so, participants were encouraged to think aloud. Completing the task took approximately 10 minutes. After the tasks, participants were interviewed using open-ended questions about their experience. The user study was audio-recorded, transcribed. The results were qualitatively analyzed according to Kuckartz [14] by two coders (with a Kappa coefficient of 0.86). We used the driving factors resulting from the DMQ as categories and, following Kuckartz, their gradual levels as subcategories. Table 1 presents our final coding system after multiple iterations.

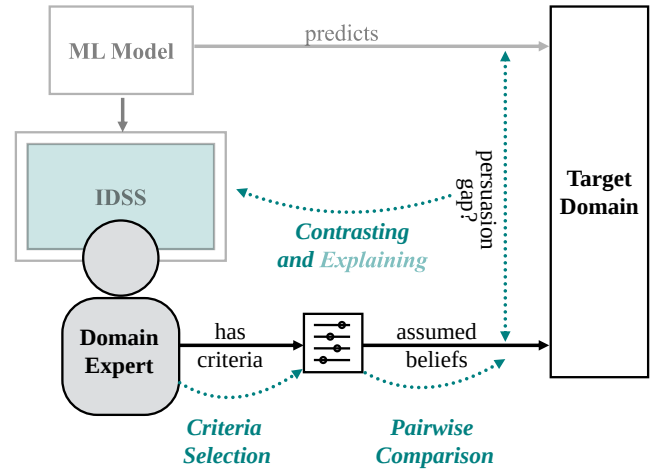


Figure 1: Explanations are triggered if there is a mismatch between the user's assumed beliefs (elicited through AHP) and the system's predictions. Blue parts relate to screens of our prototype. Muted parts relate to our proposed future work.

4 RESULTS

4.1 Interpretability Needs

The process for addendum validation was uniform for all participants. However, all participants agreed that there is no documented or formal way of deciding how to redistribute budgets. Instead, they base decisions on their personal experience and data derived from reporting features of the Alasco software. However, this approach has limitations. P2 asked for more structured workflow for addendum approval so that every stakeholder accomplishes the task in a predefined order to improve reporting. P1 would like have feedback on how well the initial budget distribution worked in

comparison to the final stage of a project. P1 and P5 asked for decision support that guides the user and recommends possible sources (e.g., based on the forecasted costs). P3 and P4 even suggested to (semi-)automate the allocation. The analysis of the DMQ indicated that control and personal involvement are important requirements for the participants. The most important subfactors were the need for *information and goals* (5 participants), *self-regulation* (4 participants), and *time/money pressure* (4 participants). It is important for the participants to have adequate and transparent data available that help them to plan, monitor, and evaluate results [7]. We leveraged these insights as guidelines for our prototype.

4.2 Prototype

We developed an IDSS interface with which participants could interact. The prototype consisted of two user flows. The first flow elicits the user's beliefs during the user onboarding through a widely accepted MADM approach. The second flow guides the user through the approval process once an addendum is requested and suggests options for budget transfer.

4.2.1 Belief Elicitation Flow. MADM approaches were used to make subjective user preferences explicit and, thus, make decision-making more transparent [21]. We leverage such an approach to elicit user beliefs about our target domain. A widely accepted and accessible MADM approach is the *analytic hierarchy process* (AHP) [10, 25]. AHP builds on a hierarchical representation of the decision problem. It leverages a user's judgments of the relative attribute importance to choose an alternative. The judgments and beliefs are elicited through pairwise comparisons of attributes. The decision criteria may be split into multiple hierarchy levels depending on the complexity. However, we limited our prototype to five decision criteria that are on the same level. We applied the wizard pattern to guide the decision maker through the steps of the AHP setup as part of a mandatory module onboarding [29]. First, users were introduced to the purpose of the flow and each step. Second, user had to select at least three criteria that they believe are important when withdrawing budget from a contract unit. Afterwards, they had to express the relative importance of each criteria through pairwise comparisons. We used the original AHP space consisting of a bidirectional Likert scale ranging from 9 (absolutely more important) to 1 (equally important) to 9. In a last step, we checked the judgments for inconsistencies and asked users to revise them if necessary. After the onboarding, users can revise their preferences anytime.

4.2.2 Intelligent Addenda Approval Flow. We enriched the manual approval flow with an intelligent overview that suggests contract units to withdraw budget from. First, the user is notified via email if a new addendum is to be reviewed. After confirming that the addendum is valid, the user sees an overview of possible contract units that may be used to accommodate the addendum. Each contract unit alternative is enriched with two types of information: (i) a score that reflects the user's beliefs. The score is calculated by AHP based on the user's relative importance of attributes as elicited during the onboarding; (ii) an intelligent suggestion that was said to take historical data into account. The suggestion may be derived through a machine learning model. Contrasting both information

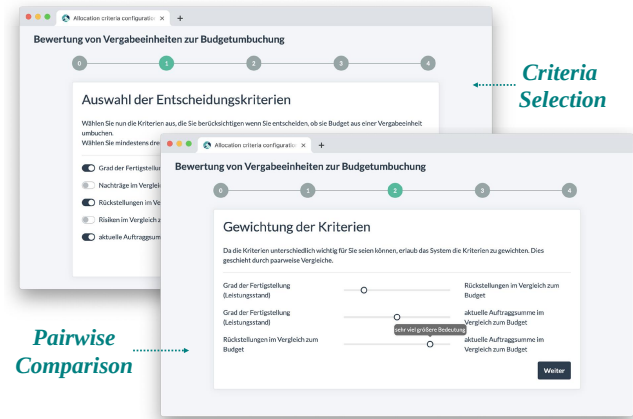


Figure 2: User's beliefs about the decision situation are elicited through AHP. In the first step, the user indicates which decision criteria are important for her. Afterwards, she compares those criteria pairwise express the relative importance.

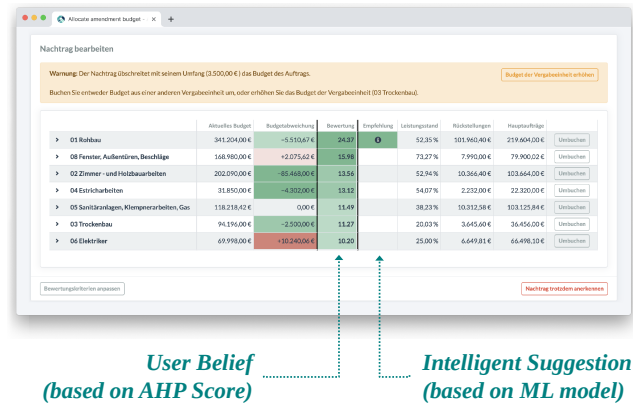


Figure 3: After an addendum is validated by the user, the IDSS gives an overview of contract units to transfer budget from. The alternatives are scored based on elicited user beliefs and contrasted with the system recommendation.

enables the user to grasp when their beliefs diverge from the system suggestion. Furthermore, it enables the system to identify and address a persuasion gap. Each column and the prediction have a tooltip that explains where the information is coming from. In our formative evaluation, the system suggestion and explanation were non-functional but based on static information. As participants were not provided with information about the underlying system logic, it resembles from a user perspective an IDSS.

4.3 Formative Evaluation

All participants were able to complete the given approval task. The results of the qualitative analysis show that all participants made

Table 1: (Left) Categories and subcategories derived from the results of the DMQ. (Right) Number of participants' statements during formative evaluation coded according to those (sub)categories.

Categories	# of Statements
Sense of Control	36
Full sense of control (<i>no doubts</i>)	12
Reinforced control (<i>feeling of guidance</i>)	8
Foreseeable behavior of the system (<i>no surprises</i>)	7
Expressed doubt/questioned system	8
Unclear statements regarding sense of control	1
Sense of Information	23
Improved experience due to information displayed	8
Satisfied with the amount of information	7
Desired additional information	7
Unclear statements regarding sense of information	1
Usability	31
Perceived increase in efficiency	7
User was hesitating/unclear	18
Expressed high mental effort	6

positive statements regarding their *sense of control* (relates to DMQ's self-regulation subfactor). 5 participants stated that their *sense of information* (relates to DMQ's information and goals subfactor) improved due to the information provided. However, 4 participants questioned the system at some point. 3 participants wished for additional information (e.g., emails or contract correspondences) or more detailed explanations (e.g., how their input affects the outcome). 4 participants perceived high mental efforts when choosing and comparing their relevant decision criteria during the onboarding. We attributed this to the fact, that they rarely had to articulate how they make addendum-related decisions before this study. However, these efforts paid off later on. 4 participants perceived increased efficiency during the addenda approval flow as they did not need to assess each alternative individually but instead could rely on the score and suggestion. Overall, we found that our prototype left the participants with an increased sense of control and information. However, the usability of the belief elicitation flow should be revised to reduce users' mental efforts. Table 1 presents a categorized summary of participants' statements.

5 LIMITATION AND FUTURE WORK

While our formative evaluation shows promising results, we acknowledge multiple limitations. Our work focuses on the limited use case of addenda approval in the construction industry. Our user studies were conducted under supervision in a controlled environment. Thus, actual user behavior and usage may be different in the field. Furthermore, our evaluation focused on the general acceptance of the approach by domain experts with a non-functional prototype. In future, we plan to conduct an experimental study that focuses on whether such an approach improves a user's understanding of an intelligent system. For this, we plan to transfer the approach to a human-grounded [8] evaluation scenario with lay users.

We believe that eliciting user beliefs and comparing them with intelligent predictions offers a promising basis for personalized explanations in XAI systems. ML algorithms take features and calculate their respective weights while optimizing a utility function. Post-hoc feature attribution methods, such as LIME [23] or SHAP [18], elicit the relative importance of a black box model's decision criteria. Similarly, decision-makers try to, explicitly or implicitly, optimize a utility function that is used to quantify their preferences regarding decision alternatives [12]. The difference is that decision-makers often do not know their utility function in advance and sometimes construct it ad-hoc during the decision-making situation. MADM methods, such as AHP, can make the user's beliefs explicit and accessible to explanation generating XAI systems. As part of our future work, we want to examine ways to relate the weights of post-hoc feature attribution methods to AHP's relative attribute importance. By this, XAI systems could adapt their explanation vocabulary (e.g., add or remove features to an explanation) or argumentation (e.g., argue with the user's expected outcome as the foil) based on the user's beliefs.

6 ACKNOWLEDGMENTS

We thank the participants who took their time to contribute their experiences and opinions from the field.

REFERENCES

- [1] Rajat Agarwal, Shankar Chandrasekaran, and Mukund Sridhar. 2016. Imaging construction's digital future. *McKinsey & Company* (2016). <https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/imaging-construction-digital-future>
- [2] David Arnott and Graham Pervan. 2012. Design Science in Decision Support Systems Research: An Assessment using the Hevner, March, Park, and Ram Guidelines. *J. AIS* 13, 11 (2012), 1. <http://aiselaisnet.org/jais/vol13/iss11/1>
- [3] Or Biran and Courtenay Cotton. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, Vol. 8, 1.
- [4] JL Blanco, S Fuchs, M Parsons, and MJ Ribeirinho. 2018. Artificial intelligence: Construction technology's next frontier| McKinsey. <https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/artificial-intelligence-construction-technologies-next-frontier>
- [5] A. Bussone, S. Stumpf, and D. O'Sullivan. 2015. The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems. In *2015 International Conference on Healthcare Informatics*. 160–169. <https://doi.org/10.1109/ICHI.2015.26>
- [6] Ajay Chander, Ramya Srinivasan, Suhas Chelian, Jun Wang, and Kanji Uchino. 2018. Working with Beliefs: AI Transparency in the Enterprise. In *Joint Proceedings of the ACM IUI 2018 Workshops co-located with the 23rd ACM Conference on Intelligent User Interfaces (ACM IUI 2018)*, Tokyo, Japan, March 11, 2018. <http://ceur-ws.org/Vol-2068/exss14.pdf>
- [7] Maria Luisa Sanz de Acedo Lizarraga, Maria Teresa Sanz de Acedo Baquedano, Maria Soria Oliver, and Antonio Closas. 2009. Development and validation of a decision-making questionnaire. *British Journal of Guidance & Counselling* 37, 3 (2009), 357–373. <https://doi.org/10.1080/03069880902956959>
- [8] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretability. *CoRR* abs/1702.08608 (2017). [arXiv:1702.08608](http://arxiv.org/abs/1702.08608)
- [9] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. 2003. The role of trust in automation reliance. *Int. J. Hum. Comput. Stud.* 58 (2003), 697–718.
- [10] Saul I. Gass. 2005. Model World: The Great Debate-MAUT Versus AHP. *Interfaces* 35, 4 (July 2005), 308–312. <https://doi.org/10.1287/inte.1050.0152>
- [11] John J. Jarvis. 1976. *Decision Support Systems: Theory*. Technical Report. Battelle Columbus Labs OH.
- [12] Ralph Keeney, Howard Raiffa, and David Rajala. 1979. Decisions with Multiple Objectives: Preferences and Value Trade-Offs. *Systems, Man and Cybernetics, IEEE Transactions on* 9 (08 1979), 403 – 403. <https://doi.org/10.1109/TSMC.1979.4310245>
- [13] Sascha Kilb and Markus Weigold. 2017. *Projektmanagement*. Springer Fachmedien Wiesbaden, Wiesbaden, 479–503. https://doi.org/10.1007/978-3-658-05368-0_20
- [14] Udo Kuckartz. 2019. *Qualitative Text Analysis: A Systematic Approach*. Springer International Publishing, Cham, 181–197. <https://doi.org/10.1007/978-3-030->

- 15636-7_8
- [15] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and Why Not Explanations Improve the Intelligibility of Context-Aware Intelligent Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (*CHI '09*). Association for Computing Machinery, New York, NY, USA, 2119–2128. <https://doi.org/10.1145/1518701.1519023>
 - [16] Robert K Lindsay, Bruce G Buchanan, Edward A Feigenbaum, and Joshua Lederberg. 1993. DENDRAL: a case study of the first expert system for scientific hypothesis formation. *Artificial intelligence* 61, 2 (1993), 209–261.
 - [17] Zachary C. Lipton. 2018. The Mythos of Model Interpretability. *Queue* 16, 3, Article 30 (June 2018), 27 pages. <https://doi.org/10.1145/3236386.3241340>
 - [18] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 4765–4774. <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions>
 - [19] Bonnie M Muir. 1994. Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics* 37, 11 (1994), 1905–1922.
 - [20] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. *CoRR* abs/1802.00682 (2018). arXiv:1802.00682 <http://arxiv.org/abs/1802.00682>
 - [21] M. Pavan and R. Todeschini. 2009. 1.19 - Multicriteria Decision-Making Methods. In *Comprehensive Chemometrics*, Steven D. Brown, Romá Tauler, and Beata Walczak (Eds.). Elsevier, Oxford, 591 – 629. <https://doi.org/10.1016/B978-044452701-1.00038-7>
 - [22] Gloria Phillips-Wren, Manuel Mora, Guiseppe A. Forgionne, Leonardo Garrido, and Jatinder N. D. Gupta. 2006. *A Multicriteria Model for the Evaluation of Intelligent Decision-making Support Systems (i-DMSS)*. Springer London, London, 3–24. https://doi.org/10.1007/1-84628-231-4_1
 - [23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) (*KDD '16*). Association for Computing Machinery, New York, NY, USA, 1135–1144. <https://doi.org/10.1145/2939672.2939778>
 - [24] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.
 - [25] Thomas L. Saaty. 2001. *Decision making for leaders : the analytic hierarchy process for decisions in a complex world* (new 3rd ed ed.). Pittsburgh, Pa., RWS Publications.
 - [26] James Schaffer, John O'Donovan, James Michaelis, Adrienne Raglin, and Tobias Höllerer. 2019. I Can Do Better than Your AI: Expertise and Explanations. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) (*IUI '19*). Association for Computing Machinery, New York, NY, USA, 240–251. <https://doi.org/10.1145/3301275.3302308>
 - [27] Edward Shortliffe. 1976. *Computer-based medical consultations: MYCIN*. Vol. 2. Elsevier.
 - [28] H.A. Simon. 1957. *Models of man: social and rational; mathematical essays on rational human behavior in society setting*. Wiley. <https://books.google.de/books?id=lEzdwwEACAAJ>
 - [29] Jenifer Tidwell. 2011. *Designing Interfaces* (Vol. 2).
 - [30] Efraim Turban, Ting-Peng Liang, and Jay E Aronson. 2005. *Decision Support Systems and Intelligent Systems: (International Edition)*. Pearson Prentice Hall.
 - [31] Stelios H. Zanakis, Anthony Solomon, Nicole Wishart, and Sandipa Dublsh. 1998. Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research* 107, 3 (1998), 507 – 529. [https://doi.org/10.1016/S0377-2217\(97\)00147-1](https://doi.org/10.1016/S0377-2217(97)00147-1)

Model Voyager: Visualization of CRF UI Models

Iyad Khaddam, Jean Vanderdonckt

Université catholique de Louvain (UCLouvain), Louvain Research Institute in Management and Organizations (LouRIM)
Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium, Belgium
{firstname.lastname}@uclouvain.be

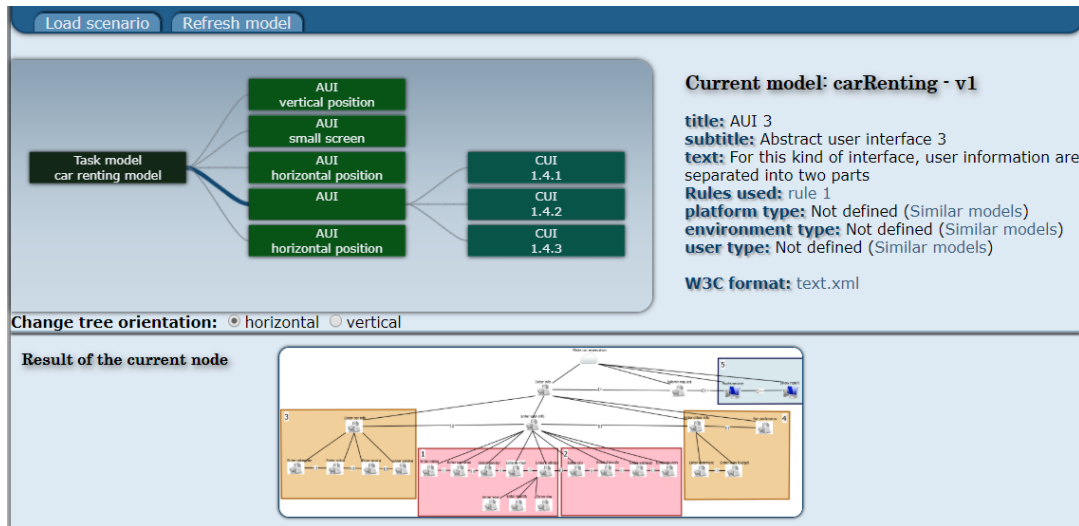


Figure 1: The Model Voyager on the “car rental” case study with hierarchy deployed.

ABSTRACT

This paper presents the Model Voyager, a web-based application for visualizing user interface models structured according to the four abstraction levels of the Cameleon Reference Framework: tasks and concepts, abstract user interface, concrete user interface, and final user interface. This application enables the designer to collect, edit, and manage collections of user interface models for a project or for maintaining an accessible catalogue of models, along with their illustrations. It also introduces three deployment mechanisms: multi-reification, multi-abstraction, and multi-translation. This paper demonstrates the applicability of the Model Voyager on the “car rental” case study, a reference example chosen by the W3C group on model-based user interfaces.

CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces**; *User studies*; *Empirical studies in interaction design*;

KEYWORDS

Cameleon Reference Framework; Design Space Exploration; Development path; Model-based user interface design; User interface models; User interface model visualization.

ACM Reference Format:

Iyad Khaddam, Jean Vanderdonckt. 2020. Model Voyager: Visualization of CRF UI Models. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398492>

1 INTRODUCTION

Model-Based User Interface Design (MB-UID) [11] consists in applying a model-based approach in order to design or develop one or many [13] user interfaces (UIs) of an interactive system by involving various models such as, but not limited to: task model, domain model, user interface model, platform model, device model, user model, help model, etc. The context of use is considered as a key aspect [3] in context-sensitive user interfaces [2], typically represented by three underlying models: a user model capturing user profiles or

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398492>

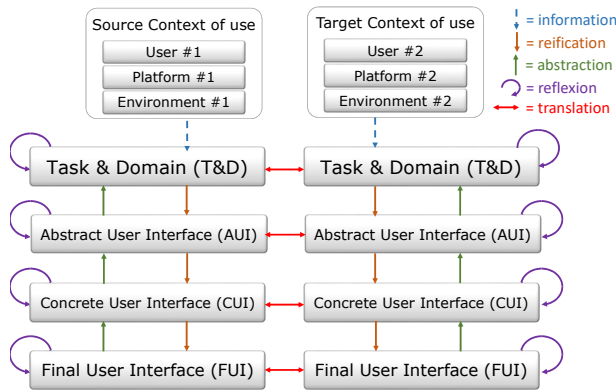


Figure 2: The Cameleon Reference Framework between two contexts of use.

clusters and related data (e.g., interaction preferences, cognitive style), a platform model capturing essential parameters potentially influencing the UI (e.g., screen resolution, interaction capabilities), and an environment model characterizing the physical and socio-organizational conditions where the user is carrying out her interactive tasks with the platform (e.g., location, level of stress, structure). The Cameleon Reference Framework (CRF) [1] structures MB-UID according to four levels of abstraction with their corresponding models as follows [10]: task and domain concepts (T&D), abstract user interface (AUI), concrete user interface (CUI), and final user interface (FUI). Having defined one context of use or more, four types of relations are used to transition among these models [10] (Fig. 2): abstraction from one level to an upward level, reification from one level to a downward level, translation for a model at the same level of abstraction from one context of use to another, and reflexion regarding a single model at the same level of abstraction.

These examples suggest that the wide variety of models involved in MB-UID, along with the traversal of relations among them, leads to a problem of grasping the various development paths [8] and understanding them. In other areas of computer science, there are plenty of software for visualizing various models and the relations among them, including for mapping, transformation, association, etc. For instance, in database management systems, a domain model obtained at the conceptual level is transformed into a logical database model at the logical level and subsequently transformed into a physical database at the physical level. Software exists that support analysts in defining and editing the meta-model associated to these models [4], in establishing and visualizing transformations among these (meta-)models, or for visualizing relations among them [17].

While several software exist for domain modelling, business process modelling, database management, workflow management, there is no counterpart for UI models. In order to fill this gap, the contributions of this paper to the area of Engineering Interactive Computing Systems are as follows:

- Model Voyager, a Web-based application for managing CRF UI models of a project based on this structure.
- A mechanism for transversal navigation among these models following vertical and horizontal relations.
- A definition of multi-reification, multi-abstraction, and multi-translation as three new mechanism for exploring multiple UI alternatives from a single starting point.
- The illustration of the Model Voyager on the “car rental”, a reference case study promoted by W3C [10].

2 RELATED WORK AND BACKGROUND

Starting from a source context of use (Fig. 2), itself represented as a user, a platform, and a corresponding environment, the CRF structures the UI development life cycle based on four levels and five relations, which are hereby summarized (see [10] for more details). When another UI needs to be developed for the same project, but for another target context of use, these levels and relations could be traversed to switch from the source context to the target context.

The *task model* is a representation of end users’ interaction with the UI and is considered as one of the models characterizing its behavior. It identifies the different types of end users and their related tasks. It is created by the task analyst to demonstrate to the different stakeholders the benefits of the application.

The *abstract user interface* (AUI) allows having a rendering of what the task model and domain model will provide [9]. The AUI is independent of any platform and environment. The AUI is composed of different abstract interaction units (AIUs) and their relationships. Two AIU categories exist:

- *Abstract containers*: represent a group of related (sub-)tasks to be conveyed in the same container, such as in the same window (for an application) or web page (for a web site). Each container can be recursively decomposed into other containers or individual components.
- *Abstract individual components*: represent individual objects that compose containers falling into four categories:
 - Input: a user can input information into the software.
 - Output: an information for the user.
 - Navigation: a transition between two containers.
 - Control: a user can control various methods of the software. In addition to these components, we find in the AUI model some relationship between the different AIU composing the interface.

The *Concrete User Interface* (CUI) is based on the AUI and concretizes it on one hand and is an abstraction of the final user interface on the other hand. It uses Concrete Interaction Units (CIU) and Concrete User Interface Relationships (CUIR). CIUs are abstractions of concrete components. Concrete components examples include buttons, progress bars

and text boxes. CIUs are not dependent of a programming language, while concrete components are. CIUs are divided into two groups: containers and individual objects. The containers are graphical components (sometimes invisible) that contain both other containers or components. Individual CIU objects cannot contain other CIUs. They allow collecting information, manipulating, changing and adapting a given model. Each programming language defines its final individual objects. Based on the target language, we can translate the CIUs into the appropriate corresponding ones on the FUI level. There are also constraints on the CIUs, such as the position, transition, alignment and adjacency. Actions can be attached to CIUs in order to generate on the FUI level.

The *final user interface* (FUI) is the last step of the forward engineering. It is the model with the lowest level of abstraction. It consists in designing the concrete model of the GUI into the chosen language of the platform [16]. At this level, the platform, the user, and the environment are all determined [9]. Still, we may take some decisions on the final design. The CRF levels are linked together by four relations:

- *Reification*: is an operation where a model is transformed into a model on a lower level of abstraction. This operation can be repeated until the FUI is reached.
- *Abstraction*: is an operation where a model is transformed into a model on a higher level of abstraction.
- *Translation*: is an operation where a model is transformed to another model at the same level of abstraction, but for a different context of use. For instance, multi-target user interfaces [1] are obtained by translation: each time another context of use is considered, a translation from the source context to the target context is triggered.
- *Reflection*: is an operation where a model is transformed into another model at the same level of abstraction, for the same context of use.

Since MB-UID could involve all these models and relations between them, many development paths are possible [8], thus making it complex and challenging to manage as they are not represented together. Some software attempt to partially address this problem. For example, *QUILL* [5] is a web based development environment displaying multiple models at the same time. Stakeholders collaboratively work on the UI design following a forward engineering MB-UID. *QUILL* visualize any involved model and relation by employing spring-based visualization techniques to represent, manipulate, and map the models and to enhance the navigability among them. For example, when a model fragment is identified, its decomposition is progressively revealed via hyperbolic trees that are expanded and collapsed depending on space available.

The Model Viewer (MoVi) [6, 7] visualizes multiple CRF models and facilitates the navigation among them. It helps in maintaining a global view of the models inside a project,

namely by propagating a change performed at one level to its subsequent levels. Similarly, *IDEALXML* [14] enables the designer to textually create, delete, and edit relations among the CRF UI models in a table. While these relations could be indeed managed, there is no graphical representation.

GEF3D [17] visualizes links between a conceptual view (UML Class diagram), an external view (a structured UI), and an external view (the final UI), thus enabling the end user to better understand how these models and their components are linked to each other, to edit them, and to see the impact. *VOYAGER 2* [4] provides a complete environment for defining meta-meta-models, meta-models, and models governed by them, as well as transformations at these three levels of abstraction. This environment mainly focuses on the domain model and does not cover UI models. *MEGA-UI* [15] is an environment helping analysts to manage all model perspectives (*i.e.*, model, meta-model, transformation) at various CRF levels. It allows to navigate between components in the same model and navigate from one model to another model only in the same system. It does not present the external models from another system that can contribute to the whole design.

Therefore, we observe that available tools are mainly aimed at editing CRF UI models in order to support the UI development life cycle, which is normal. There is no genuine software for visualizing the various models and their relations for a single project, for one or multiple contexts of use, which is precisely the goal of this work.

3 MODEL VOYAGER

The Model Voyager^a consists of a web application enabling any user to create a new project and to feed it with any CRF UI model required to support its development life cycle and for establishing relations among them. For each model, the visitor can download the model information in an XML format. A graphical schema is displayed on the web site and can be downloaded as well. If the FUI model source code is available, it can be downloaded too. The Model Voyager provides two main functionalities for visitors:

- Display model information: any model is in the Model Voyager according to simple data structure that is automatically displayed when selected.
- Navigate through levels: models involved in a single project are displayed in a dynamic tree that can be expanded and collapsed depending on model navigation.

Displaying Model Information

The Model Voyager stores a project or case study as a hierarchy of directories, which can be recursively decomposed into sub-directories. Each directory contains one CRF model stored in an *info.xml* file with the following descriptive data:

^aFreely accessible at <https://sites.uclouvain.be/mbui/index.php?t=carRenting/v1>

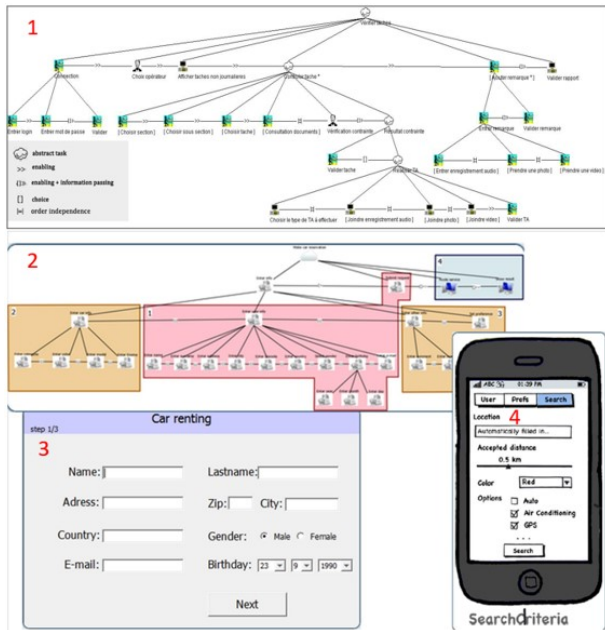


Figure 3: Screenshots for different models: (1) The task, (2) the AUI containers, (3) the CUI and (4) the FUI.

- title: the title of the current node corresponding to one CRF level. For example, “Task model for renting a car”.
- subtitle: a more detailed header than the title. For example, “This task model represents the expected way to rent a car via the company”.
- text: the explanation of the current node. For example, a textual description of a scenario covered by the task.
- rules used: the different rules used to achieve this model, if any. Each rule can be expressed in natural language, structured query language, or via a link to the web site where all transformation rules have been defined. For example, a set of rules used for deriving an AUI from the “car rental” task model are: change of the size of the interface (`sourceInterface.size > outputInterface.size`), groups different AIUs by category (e.g., personal information, car information, additional information, and result), decompose the “Enter birthdate” task into three sub-tasks: “Enter year”, “Enter month”, and “Enter day”.
- user type: link to user model in force at this level. For example, link to “Tourist model”.
- platform type: link to the platform model in force at this level. For example, link to “Smartphone model”.
- environment type: link to the environment model in force at this level. For example, link to “Stressing place”.

In addition to the `info.xml` file, we can add screenshots for the model. We can also add a URL to any external representation of the model. For example, Fig. 3 contains: one for a task model, one for the AUI containers (on the task model), one for the CUI and finally one for the FUI.

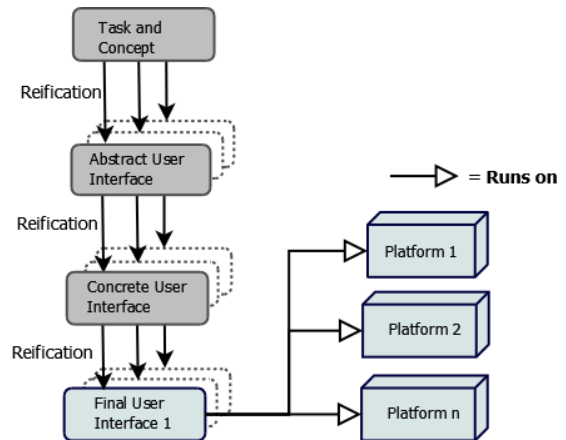


Figure 4: Example of multi-reification at the FUI level.
Navigate through Levels

Forward engineering in MB-UID basically exploits the reification relation from one level to any subsequent level: task to AUI, AUI to CUI, and CUI to FUI. Since this process can be repeated at any level, it gives rise to a tree structure, where each node holds a particular CRF model derived from the previous one. Each tree node holds a particular CRF model stored via its `INFO.XML` and `IMAGE.PNG` files: when a node is selected, the dynamic tree expands the sub-nodes to enable the visitor to drill down the models. When this node is selected again, the dynamic tree collapses its sub-graph to let the visitor moving up and down through the tree quickly, while being informed on the current position. When a particular node is activated, detailed information on the model in this node are displayed.

Implementation

To realize this dynamic tree, a PHP script parses the hierarchy of (sub-)directories and its contained files to automatically generate a JSON file used as input to the dynamic tree, rendered by the JavaScript InfoVis Toolkit^b. Adding a new model at any level simply consists in adding a new directory at the right place in the hierarchy. The Model Voyager updates its visualization by re-running the script each time this structure is modified. When a node is selected, a JavaScript executes an Ajax request on `info.xml` to display its contents and establish the links following the relations.

Finding Similar Models

For the three types user, platform, and environment, the Model Voyager displays the link “Similar models” next to it to find similar models with related contexts of use. This link opens a new window (Fig. 6-c): the visitor browses different variations of the model per context of use. Similarity is defined dynamically, based on existing models within the tool. When two variations co-exist for the same context of use,

^bSee <https://phillogb.github.io/jit/>

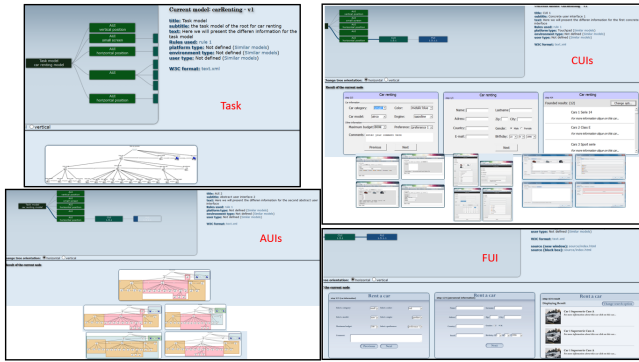


Figure 5: The “Car rental” models in the Model Voyager. a drop-down list filters the alternatives by context of use. When multi-target UIs are devised in a project, reification is repeated several times from a same starting point. In order to reduce this repetition for all alternatives, we define a new notion aimed at defining multi-target UIs at once: a *multi-reification* consists in applying a synchronized reification at a given CRF level of abstraction for the same project, but for different contexts of use, and to semantically link them. For example, Fig. 4 graphically depicts a multi-reification performed at the CUI level for different target platforms: instead of creating a single reification for each alternative, the designer specifies the desired variations of the context of use (here, three) and applies a multi-reification at once. A multi-reification can be applied at any non-terminal CRF level. Similarly, a *multi-abstraction* consists in applying a synchronized abstraction at a given CRF level of abstraction for the same project, but for different contexts of use. A multi-abstraction can be applied at any non-initial CRF level. A multi-translation applies a synchronized translation at any level for many targets. Model Voyager supports multi-reification by allowing to navigate among these models once the parent model is activated.

Scenarios

The Model Voyager browses the directory of one project or case study at a time, which we call the scenario directory. One can visit another scenario by clicking on the “Load scenario” link on the top-left corner of the main page (Fig. 6-d).

It is possible to add anew scenarios but a user account on the server is required. Users can collaborate by enriching the content of a scenario with different models, and thus collaborate to foster the exploration of the design space. Instructions for this purpose are publicly available^c.

4 A CASE STUDY: THE CAR RENTAL

Among others, the “car rental case study is a reference case study studied in the frame of the W3C Charter Group on MB-UID^d. We will therefore consider it as a reference example to

illustrate the capabilities of the Model Voyager. Particularly useful are also those navigations offered by the Model Voyager to traverse various levels of abstraction or at the same level of abstraction, but also to travel along dimensions of the context of use, and thus answering the question: if we maintain the same CRF level of abstraction, what is the consequence of changing the user model, the platform and/or the environment?

In the Model Voyager, one chooses the “Car Rental” scenario and the version to visit. The first stop in the walk-through is the task model. The task model is displayed in the appropriate section (Fig. 5-Task), using the CTTE notation [12]. On the right side, supplementary information on the task model is displayed. To see how the task model changes on a different platform, the link “Similar Model” should be selected on the same line of “platform type”. This will open variations of the task model for different platforms. The same process is valid for the user and the environment. Activating the task model node opens the sub-nodes for the AUIs (Fig. 5-AUIs). In this scenario, five nodes are available representing five AUI variations: two for vertical layouts for smartphones, one for horizontal layout for a landscape tablet, one for small screens and finally a generic one. Selecting any node shows how the distribution of tasks in terms on AIUs is made. On the right side, supplementary information on the AUI model is displayed. Other variations of the AUI model based on the current context of use or for another one are accessible by browsing the “Similar models” link.

Activating an AUI model node opens the sub-nodes for the CUIs (Fig. 5-CUIs). For instance, clicking on the generic AUI node displays the three variations of CUI models. Clicking on any of the CUI model nodes deploys the CUI containers. One remarkable difference is the existence of several CUI models: at the CUI level, we are getting closer to the final UI and the widgets start to take their shapes. On the right side, supplementary information on the CUI model is displayed. The same processes applies for the CUI model concerning adaptation of the model to any context of use.

Activating a CUI model node opens the sub-nodes for the FUIs. For instance, click on the AUI horizontal position, then the CUI models, a node for the FUI appears (Fig. 5-FUI). Selecting a FUI node shows how a CUI is reified into a FUI.

5 CONCLUSION

We presented the Model voyager, a web application managing and visualizing UI models and relations structured according to the Cameleon Reference Framework (CRF). We explained the functionalities of the software, the models presented, and information on these models.

We demonstrated the use of the tool to browse a model through the “Car rental” case study. The goal of Model Voyager is to allow visitors to visualize UI models according to

^cSee <http://sites.uclouvain.be/mbui/guide.html>

^dSee use case UC1 in <https://www.w3.org/TR/mbui-intro/>

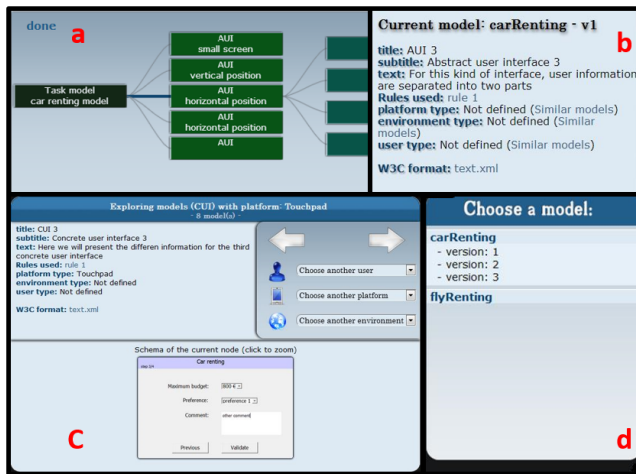


Figure 6: The tool screens: (a) the navigator, (b) model information, (c) similar models and (d) model selection.

the four levels of the CRF. Reification and abstraction relations are supported in the dynamic tree, while reflexion and translation are supported through the “Similar models” link.

The main contributed value of Model Voyager is to visually guide the visitor among the multiple CRF UI models, to explore design alternatives at any level. This approach fosters a design space exploration that could increase the observability and browsability of a same case study, especially when considering different contexts of use. This design space exploration is expected to become useful for practitioners like analysts, designers, and developers who are responsible for managing several UI models, but also for teachers to facilitate understanding the usage of the CRF by students.

We intend to turn Model Voyager into a shared reference for the MB-UID community at large. We are looking after more case studies and contents. This is an open invitation to all interested parties to contact us in order to create accounts for them to share their contents or to ask us to install an instance of the tool on their own server.

REFERENCES

- [1] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (6 2003), 289–308. [https://doi.org/10.1016/S0953-5438\(03\)00010-9](https://doi.org/10.1016/S0953-5438(03)00010-9)
- [2] José Creissac Campos and F. Martins. 1996. Context Sensitive User Interfaces. In *Formal Aspects of the Human Computer Interface* (electronic Workshops in Computing), C. Roast and J. Siddiqi (Eds.). Springer-Verlag London, Sheffield Hallam University, UK. <http://www.springer.co.uk/eWiC/Workshops/FACHI/>
- [3] Joëlle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. 2005. Context is Key. *Commun. ACM* 48, 3 (March 2005), 49–53. <https://doi.org/10.1145/1047671.1047703>
- [4] Vincent Englebert and Jean-Luc Hainaut. 1999. DB-MAIN: A Next Generation Meta-CASE. *Inf. Syst.* 24, 2 (1999), 99–112. [https://doi.org/10.1016/S0306-4379\(99\)00007-1](https://doi.org/10.1016/S0306-4379(99)00007-1)
- [5] Vivian Genaro Motti, Dave Raggett, Sascha Van Cauwelaert, and Jean Vanderdonckt. 2013. Simplifying the Development of Cross-platform Web User Interfaces by Collaborative Model-based Design. In *Proceedings of the 31st ACM International Conference on Design of Communication* (September 30–October 01, 2013) (SIGDOC '13). ACM, New York, NY, USA, 55–64. <https://doi.org/10.1145/2507065.2507067>
- [6] Mufida Mir'atul Khusna, Gaëlle Calvary, Sophie Dupuy-Chessa, and Yann Laurillau. 2015. MoVi: models visualization for mastering complexity in model driven engineering. In *Proceedings of the 2015 British HCI Conference, Lincoln, United Kingdom, July 13-17, 2015*, Shaun W. Lawson and Patrick Dickinson (Eds.). ACM, 281–282. <https://doi.org/10.1145/2783446.2783611>
- [7] Mufida Mir'atul Khusna, Sophie Dupuy-Chessa, and Gaëlle Calvary. 2016. Mastering Model Driven Engineering complexity by interactive visualization. *Technique et Science Informatiques* 35, 2 (2016), 175–202. <https://doi.org/10.3166/tsi.35.175-202>
- [8] Quentin Limbourg and Jean Vanderdonckt. 2009. Multipath Transformational Development of User Interfaces with Graph Transformations. In *Human-Centered Software Engineering - Software Engineering Models, Patterns and Architectures for HCI*, Ahmed Sefah, Jean Vanderdonckt, and Michel C. Desmarais (Eds.). Springer, 107–138. https://doi.org/10.1007/978-1-84800-907-3_6
- [9] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, and Murielle Florins. 2004. USIXML: A User Interface Description Language Supporting Multiple Levels of Independence. In *Proceedings of Workshops in connection with the 4th International Conference on Web Engineering (ICWE '04). Engineering Advanced Web Applications* (28-30 July, 2004) (DIWE '04), Maristella Matera and Sara Comai (Eds.). Rinton Press, 325–338.
- [10] Gerrit Meixner, Gaëlle Calvary, and Joëlle Coutaz. 2014. Introduction to Model-Based User Interfaces - W3C Working Group Note. <https://www.w3.org/TR/mbui-intro/>
- [11] Gerrit Meixner, Fabio Paternò, and Jean Vanderdonckt. 2011. Past, Present, and Future of Model-Based User Interface Development. *i-com Zeitschrift für interaktive und kooperative Medien* 10, 3 (2011), 2–11. <https://doi.org/10.1524/icom.2011.0026>
- [12] G. Mori, F. Paterno, and C. Santoro. 2002. CTTE: support for developing and analyzing task models for interactive system design. *IEEE Transactions on Software Engineering* 28, 8 (2002), 797–813.
- [13] Fabio Paternò and Carmen Santoro. 2002. One Model, Many Interfaces. In *Computer-Aided Design of User Interfaces III, Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces, May, 15-17, 2002, Valenciennes, France*, Christophe Kolski and Jean Vanderdonckt (Eds.). Kluwer, 143–154.
- [14] Francisco Montero Simarro and Víctor López-Jaquero. 2006. IdealXml: An Interaction Design Tool. In *Computer-Aided Design of User Interfaces V, Proceedings of the Sixth International Conference on Computer-Aided Design of User Interfaces, CADUI 2006 6-8 June 2006, Bucharest, Romania*, Gaëlle Calvary, Costin Pribeanu, Giuseppe Santucci, and Jean Vanderdonckt (Eds.). Springer, 245–252. https://doi.org/10.1007/978-1-4020-5820-2_20
- [15] Jean-Sébastien Sottet, Gaëlle Calvary, Jean-Marie Favre, and Joëlle Coutaz. 2009. Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces: MEGA-UI. Springer London, London, 173–200. https://doi.org/10.1007/978-1-84800-907-3_8
- [16] Jean Vanderdonckt. 2005. A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In *Advanced Information Systems Engineering*, Oscar Pastor and João Falcão e Cunha (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 16–31.
- [17] Jens von Pilgrim and Kristian Duske. 2008. Gef3D: A Framework for Two-, Two-and-a-Half-, and Three-Dimensional Graphical Editors. In *Proceedings of the 4th ACM Symposium on Software Visualization (SoftVis '08)*. Association for Computing Machinery, New York, NY, USA, 95–104. <https://doi.org/10.1145/1409720.1409737>

Computing Aesthetics of Concrete User Interfaces

Patrick Mbenza, Nicolas Burny

Université catholique de Louvain (UCLouvain), Louvain Research Institute in Management and Organizations (LouRIM)
Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium – {patrick.mbenza, nicolas.burny}@uclouvain.be, Belgium

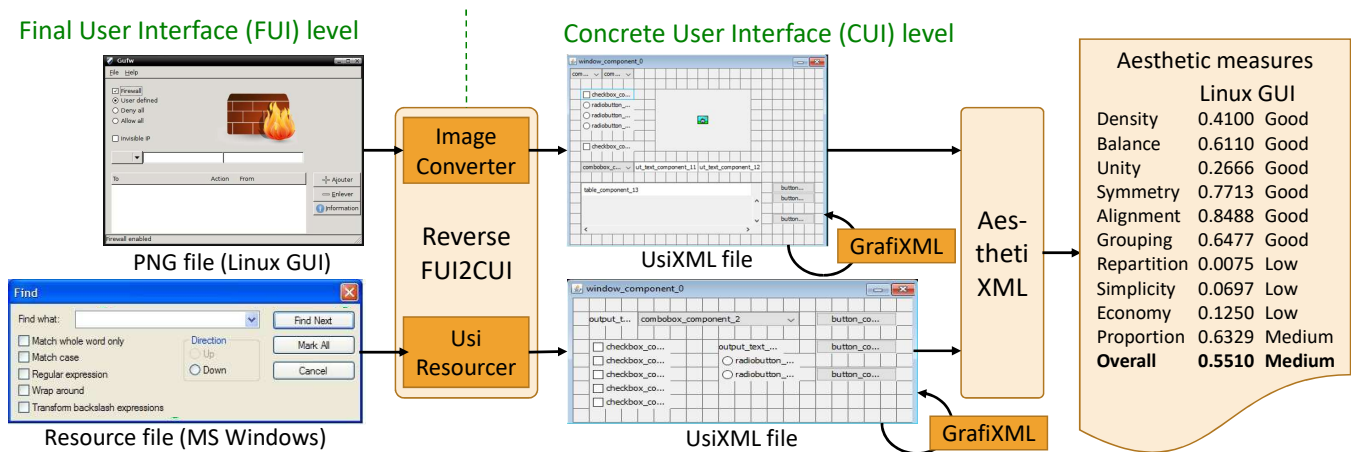


Figure 1: Process for computing aesthetics: reverse engineering, editing, and computation of aesthetic measures.

ABSTRACT

Since aesthetics of a graphical user interface have become a sub-factor of the ISO 25010 standard on software quality, organisations start wondering how to practically evaluate this property in a consistent way. To address this challenge, we present a process for computing aesthetics at the level of a Concrete User Interface instead of Final User Interface to make it platform independent. This process consists of performing the following steps: reverse engineer a final user interface into its concrete equivalent, optionally edit it, and automatically compute ten aesthetic metrics at the level of concrete user interfaces.

CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces**; *User studies*; *Empirical studies in interaction design*;

Permission to make digital or hard copies of all or part 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 permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398493>

KEYWORDS

Aesthetics, metrics/measures, usability testing

ACM Reference Format:

Patrick Mbenza, Nicolas Burny. 2020. Computing Aesthetics of Concrete User Interfaces. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3393672.3398493>

1 INTRODUCTION AND BACKGROUND

Since more than a decade, a wave of research has emerged suggesting that aesthetics is a strong determinant of user satisfaction, and that a positive correlation exists between aesthetic quality of a Graphical User Interface (GUI) and its perceived usability [20]. In addition, GUI aesthetics is now considered as a sub-factor of Usability”, one of the seven factors of the ISO 25010 standard on software quality. Consequently, organizations and evaluators start thinking how to practically assess this sub-factor, namely for reproducing it and, perhaps, for qualifying the ISO certification. Practitioners are particularly concerned on how to understand aesthetics, how to evaluate it and how to control it. A practical procedure for evaluating GUI aesthetics does not exist yet, like those found in Web accessibility where precise checkpoints are defined and consistently assessed.

Popular wisdom is generally skeptical about the fact that aesthetics can be effectively measured, and most people believe that aesthetic appeal is primarily a subjective judgment

that is not based on any notion of purpose. This purely intuitive conviction is the equivalent of what is called *subjectivism* in philosophy, which is summed up by the famous sentence “beauty is in the eye of the beholder”. The popular belief is also akin to *relativism*, which claims that the culture to which a person belongs or the context in which they find themselves shape their sense of aesthetics and are therefore not separated from it. The opposite philosophical view, *objectivism*, holds that the aesthetic qualities belong to objects themselves, as do other physical attributes, and are independent of the valuating subject. Since aesthetic experience depends on such qualities, it gives us some knowledge: our aesthetic judgment about an object is correct whenever it conforms to the aesthetic characteristics the object possesses.

Thus, there are probably universal patterns of beauty, interpreted as aesthetic qualities valid for all humans in every culture, and all humans evaluate objects that correspond to these patterns as beautiful, provided that they can perceive these patterns in the same way. Evaluating these patterns should therefore be feasible by measuring quantitative *metrics* (or *measures*, a term preferred in the field of software testing) representing aesthetics^a. They are not thought of as arbitrary by most people, making the line between those standards and universal features of beauty blurry and difficult to define. So regardless of our philosophical world view, the idea of using metrics to measure aesthetic quality remains relevant, as many have believed throughout history (for instance, Euclide’s *Elements* -300 B.C. contain the oldest known definition of the famous *golden number*). Several metrics have been defined for this purpose attempting to capture similar or different, if not overlapping and intertwined, aspects of aesthetics, such as [5, 14, 16, 18]. Computing these metrics manually on a GUI is tedious and error prone process, thus justifying why several software have been developed to support this activity [8, 14, 24], environments to compute them in an efficient way [1–3], and integration in model-driven engineering [7].

While all these tools are useful, they share a common restriction: they work on a particular GUI format that is depending on the operating system and the computing platform running the GUI under evaluation: a HTML page, a screenshot captured as a JPG file, a text file, etc. In order to render this process as autonomous as possible with respect to peculiarities imposed by the computing platforms and these operating systems, this paper investigates how computing aesthetics could be achieved at the Concrete User Interface (CUI) level (Fig. 1) instead at the Final User Interface (FUI) level, as defined in the CRF [4], thus making it independent of the platform and its operating system and format.

^aEven in a subjectivist or relativist world view, there are still beauty standards” that prevail in society for a given period of time

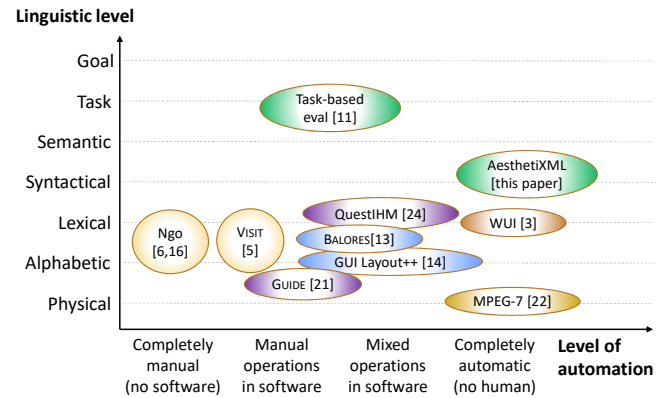


Figure 2: Classification of some software by linguistic level.

2 RELATED WORK

Since the dawn of Character-based User Interfaces, researchers have been interested in computing metrics of a screen, primarily by calculating the value of visual techniques [23]: Tullis introduced the first formula for computing density [21], Streveler & Wasserman defined what symmetry and balance mean [10]. When GUIs appeared, so did the interest to estimate their aesthetic quality [20]. Approaches to computing GUI aesthetics can be classified according to Nielsen’s linguistic level of interaction [17] (Fig. 2):

- *Physical* approaches treat a GUI as a pure image whose light, colors, contrast, and other physical features are computed to estimate its aesthetics. For example, Uribe *et al.* [22] extract descriptors from the MPEG-7 vector of a PNG file to compute a metric correlated to the perceived usability. This low-level does not permit any interpretation or explanation of the results.
- *Alphabetic* approaches consider the alphabet of characters to compute metrics, as in GUIDE [21] or in [10].
- *Lexical* approaches interpret a GUI as an image where the elements are subject to some computation. For example, GUI LAYOUT++ does this job on screen regions of different types on a GUI mockup, GUI widgets are treated as screen areas for computing consistency [14], complexity for mobile phones [19], or a set of metrics, e.g., VISIT [5], BALORES [13], QUESTIM [24].
- *Syntactical* approaches rely on a GUI definition in terms of layout, behavior, and contents to characterize their distribution. For example, the DOM of a web page is parsed to detect its GUI elements, such as controls, menus, banners, logos, and contents. Static and dynamic analyses of this file simulate the functioning of the GUI to compute some metrics in WUI [3].
- *Task-based* approaches connect the GUI with a task model in order to compute metrics, such as task effectiveness, efficiency, and distribution [11].

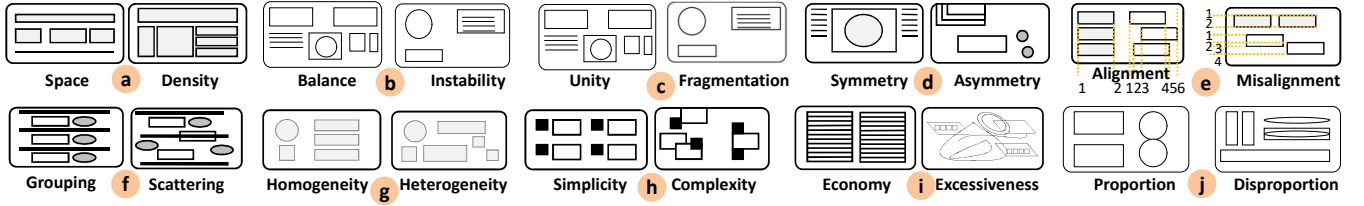


Figure 3: The ten measures for computing aesthetics of a concrete user interface.

Actually, physical and lexical approaches benefit from some form of platform independence since they represent a GUI as a matrix of pixel and apply pixel-based methods to compute their metrics. Physical approaches are also independent of the format: a screenshot, a wireframe, a picture, an actual photograph or a mockup are all admissible input formats in principle. Some environments also appear to integrate aesthetic computation in GUI design, such as ISO/IEC 15939 standard compliance for aesthetics [2], or in model-based design [7]. Fig. 2 also sorts tools by their level of automation, ranging from a completely manual process [16] to a completely automated process [3]. Between these extremes, operations could be decided by the evaluator and performed by the software or subject to mixed-initiative. This paper raises the level of abstraction of aesthetic computation with respect to these aforementioned levels: from alphabetic to lexical.

3 METHOD AND METRICS

Our research method for collecting formula for aesthetic metrics consisted in browsing the literature to find the existing definitions and in selecting among them a minimal subset that captures the aesthetic quality through properties. We also modified some formulas when necessary or introduced new ones where there were none. In the end, we want to test the universality of these metrics with human subjects.

So, first of all, we compiled and consolidated metrics defined in [5, 6, 9, 14, 16, 18, 23], which are references relying on *Visual Design*. Visual Design originates from both GUI Design and Graphic Design. It aims to shape and improve user experience through the arrangement of information items such as text, images, diagrams, tables in such a way that the whole experience is visually appealing, insightful and easily understandable. And it does this by utilizing an assortment of principles, including unity, emphasis, spacing, hierarchy, balance, contrast, dominance, and similarity. Visual Design additionally considers the effects of illustrations, photography, typography, layouts, and color on the usability of products and their aesthetic appeal. Second, we reduced the pool of metrics by dispensing those that intuitively seemed extraneous, redundant, or exclusively related to usability rather than aesthetics, requiring complex task analysis to be computed accurately, or based on a visual feeling and impression, and therefore too complex to implement.

Density, as opposed to space (Fig. 3a), is defined as [...]the extent to which the screen is covered with objects [6]:

$$DM = 1 - \frac{\sum_{i=1}^n a_i}{a_{\text{frame}}} \in [0, 1] \quad (1)$$

where a_i and a_{frame} are the areas of object i and the frame, and n is the number of objects on the frame.

Balance, as opposed to instability (Fig. 3b), can be defined as [...]a search for equilibrium along a vertical or horizontal axis in the layout [23]:

$$BM = 1 - \frac{\left| \frac{W_l - W_r}{\max(|W_l|, |W_r|)} \right| + \left| \frac{W_t - W_b}{\max(|W_t|, |W_b|)} \right|}{2} \in [0, 1] \quad \text{with}$$

$$w_j = \sum_i d_{ij} \left(\frac{a_{ij}}{a_{\text{max}}} + |c_{ij} - c_{\text{frame}}| + s_{ij} \right), 0 \leq c_{ij}, c_{\text{frame}}, s_{ij} \leq 1$$

where $j=L, R, T, N$ for left, right, top, and bottom, respectively; a_{ij} , c_{ij} , and s_{ij} are the area, colour, and shape of object i on side j ; d_{ij} is the distance between the central lines of the object and the frame; and n_j is the total number of objects on the side. Each colour has a weight between 0 (white) and 1 (black). **Unity** is coherence [6], as opposed to fragmentation (Fig. 3c), the design and arrangement of the elements in such a way that they seem to belong together.

$$UM = \frac{|UM_{\text{form}}| + |UM_{\text{space}}|}{2} \in [0, 1]$$

$$UM_{\text{form}} = 1 - \frac{n_{\text{color}} + n_{\text{shape}} + n_{\text{size}} - 3}{3n}$$

$$UM_{\text{space}_i} = 1 - \frac{a_{\text{layout}} - \sum_i^n a_i}{a_{\text{frame}} - \sum_i^n a_i}$$

Symmetry, as opposed to asymmetry (Fig. 3d), is the duplication of features along horizontal, vertical axes [5]:

$$SYM = 1 - \frac{|SYM_{\text{vertical}}| + |SYM_{\text{horizontal}}| + |SYM_{\text{radial}}|}{3} \in [0, 1]$$

$$SYM_{\text{vertical}} =$$

$$\begin{aligned} & |X'_{UL} - X'_{UR}| + |X'_{LL} - X'_{LR}| + |Y'_{UL} - Y'_{UR}| + |Y'_{LL} - Y'_{LR}| + \\ & |H'_{UL} - H'_{UR}| + |H'_{LL} - H'_{LR}| + |B'_{UL} - B'_{UR}| + |B'_{LL} - B'_{LR}| + \\ & |\Theta'_{UL} - \Theta'_{UR}| + |\Theta'_{LL} - \Theta'_{LR}| + |R'_{UL} - R'_{UR}| + |R'_{LL} - R'_{LR}| \end{aligned}$$

12

$SYM_{\text{horizontal}}$ = same as SYM_{vertical} , but swap LL and UR

SYM_{radial} = same as $SYM_{\text{horizontal}}$, but swap LL and LR

where $X'_j, Y'_j, H'_j, B'_j, Q'_j$, and R'_j are, respectively, the normalised values of X_j, Y_j, H_j, B_j, Q_j , and R_j with

$$X_j = \sum_i^{n_j} |x_{ij} - x_c|, Y_j = \sum_i^{n_j} |y_{ij} - y_c|, H_j = \sum_i^{n_j} h_{ij}, B_j = \sum_i^{n_j} b_{ij}$$

$$R_j = \sum_i^{n_j} \sqrt{(x_{ij} - x_c)^2 + (y_{ij} - y_c)^2}, \Theta_j = \sum_i^{n_j} \left| \frac{y_{ij} - y_c}{x_{ij} - x_c} \right|$$

where $j=UL, UR, LL, LR$ stand for upper-left, upper-right, lower-left, and lower-right, respectively; (x_{ij}, y_{ij}) and (x_c, y_c) are the co-ordinates of the centres of object i on quadrant j and the frame; b_{ij} and h_{ij} are the width and height of the object; and n_j is the total number of objects on the quadrant.

Alignment, as opposed to misalignment (Fig. 3e), is guaranteed if the number of vertical alignment points in a row and the number of horizontal alignment points in a column is [...] minimized [23]. We proposed an original formula to measure alignment:

$$AM = \frac{4n - (n_{\text{hap}} * n_{\text{vap}})}{4n - 4} \in [0, 1]$$

where n_{vap} and n_{hap} are the numbers of vertical and horizontal alignment points and n is the number of objects on the frame. **Grouping**, as opposed to scattering (Fig. 3f), means using similar sizes, shapes, colors, or positions for related information and keeping them close to each other. It promotes unity, but unity is achievable without it. We proposed an original formula to measure grouping considered as spatial unity and alignment for elements of the same type:

$$GM = \frac{\sum_i^n 0.85 \times UM_{\text{space}_i} + 0.15 \times AL_i}{n} \in [0, 1] \text{ with}$$

$$UM_{\text{space}_i} = 1 - \frac{a_{\text{layout}} - \sum_j^n a_{ij}}{a_{\text{frame}} - \sum_j^n a_{ij}}, AL_i = \min(1, \frac{4n_i - \frac{n_{\text{hap}} \times n_{\text{vap}}}{n_i}}{4n_i - 4})$$

where a_{layout} , a_{frame} , and a_{ij} are the areas of the layout, of the frame, and of object j of type i ; n_i is the number of objects of type i ; n_{hap} and n_{vap} are the number of horizontal and vertical alignment points for objects of type i . **Homogeneity**, as opposed to heterogeneity (Fig. 3g), is a measure of how evenly the objects are distributed among the quadrant [6]:

$$HM = \frac{\left(\frac{n}{4}\right)^4}{n_{UL}!n_{UR}!n_{LL}!n_{LR}!} \in [0, 1]$$

where $n_{UL}, n_{UR}, n_{LL}, n_{LR}$ are the numbers of objects on the upper-left, upper-right, lower-left, and lower-right quadrants, respectively; n is the number of objects on the frame, where a_i and a_{frame} are the areas of object i and the frame; and n is the number of objects on the frame or GUI. For homogeneity, our tests have shown that the measurement would be more precise if it depended on the areas of the elements rather than on their number. **Simplicity**, as opposed to complexity (Fig. 3h), is directness and singleness

of form, a combination of elements that results in ease in understanding the meaning of a pattern [6]:

$$SM = \frac{3}{n + n_{\text{hap}} + n_{\text{vap}}} \in [0, 1]$$

where n_{vap} and n_{hap} are the numbers of vertical and horizontal alignment points; n is the number of objects on the frame. **Economy**, as opposed to excessiveness (Fig. 3i), is the careful and discreet use of display elements to get the message across as simply as possible. Economy is achieved by using as few styles, formats, and colors as possible [6]:

$$EM = \frac{3}{n_{\text{color}} + n_{\text{shape}} + n_{\text{size}}} \in [0, 1]$$

where n_{color} and n_{shape} and n_{size} are the numbers of sizes, colours, and shapes used, respectively. **Proportion**, as opposed to disproportion (Fig. 3j), is the principle that [...] strives for an aesthetically appealing ratio between the dimensions of interaction objects [23]:

$$PM = \frac{|PM_{\text{object}}| + |PM_{\text{layout}}|}{2} \in [0, 1]$$

$$PM_{\text{object}} = \frac{1}{n} \sum_i^n \left[1 - \frac{\min(|p_j - p_i|, j = sq, r2, gr, r3, ds)}{0.5} \right]$$

$$PM_{\text{layout}} = 1 - \frac{\min(|p_j - p_{\text{layout}}|, j = sq, r2, gr, r3, ds)}{0.5} \text{ with}$$

$$p_{sq}, p_{r2}, p_{gr}, p_{r3}, p_{ds} = \frac{1}{1}, \frac{1}{1.414}, \frac{1}{1.618}, \frac{1}{1.732}, \frac{1}{2}$$

$$p_i = \begin{cases} r_i & r \leq 1 \\ \frac{1}{r_i} & \text{otherwise} \end{cases} \quad p_{\text{layout}} = \begin{cases} n_{\text{layout}} & r \leq 1 \\ \frac{1}{r_{\text{layout}}} & \text{otherwise} \end{cases}$$

$$r_i = \frac{h_i}{b_i} \quad r_{\text{layout}} = \frac{h_{\text{layout}}}{b_{\text{layout}}}$$

where b_i and h_i are the width and the height of the object i ; b_{layout} and h_{layout} are the width and height of the layout; and p_j is the proportion of shape j .

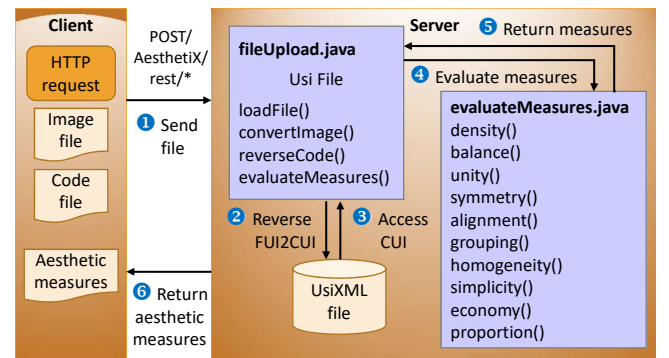


Figure 4: Web Service (API) for aesthetic computation.

4 SOFTWARE FOR COMPUTING CUI AESTHETICS

We implemented in Java the ten above metrics in a RESTful web service to facilitate its integration into other applications and to foster reusability. The software architecture (Fig. 4) is that of a typical web service accessible via a URI that the client uses to send a GUI either as an image file, e.g., a PNG, a JPG or a GIF file, or as Windows RC resource file. The server converts the input file into a UsiXML CUI file [12], either by image processing or by reverse engineering of the resource file [25]. It then calculates the ten metrics on the Concrete User Interface and returns the scores to the client in HTML format (e.g., as in Fig. 1) or returns an error message if the UsiXML file is not well-formed. In this case or for editing the UsiXML file, a graphical editor can be used [15].

5 USER STUDY

We conducted a user study to evaluate the system and the metrics. We selected three very different GUIs:

- (1) a GUI with low symmetry or balance assessments.
- (2) a GUI looking cluttered and likely to have density, simplicity, economy, grouping, and alignment issues.
- (3) a GUI seeming flawless and likely to receive high scores for most metrics.

We then created duplicates of the first two interfaces on which we rectified the apparent flaws. It resulted in a final sample of five interfaces (fig. 5) that we submitted to the automatic computation of the metrics, which ranked them according to an overall score consisting of a weighted average of all the metrics. Afterward, we gathered a heterogeneous group of sixteen users, eight males, and eight females of diverse ethnic/cultural backgrounds and age groups. We asked them to assess the aesthetic quality of the same five interfaces on a “low”, “medium”, or “high” scale, and to explain their choices. Only six of the participants had some familiarity with the fields of Information Technology, and only three knew about

visual/interface design concepts. Participants classified the five GUIs in the order in which they were chosen to be classified. This concordance in the hierarchy seems to confirm that some essential aspects of aesthetics are relatively objective. Or that they lean on standards so universally adopted that trying to measure them is not non-sensical.

Due to lack of space, we provide the table of results as a supplemental resource. Despite a small loss of precision due to the conversion to CUI, the program also rated them in the same order. But only for some weight values in the calculation of the weighted average. e.g.: 0.005 for *density*, 0.35 for *balance*, 0.03 for *unity*, 0.03 for *symmetry*, 0.03 for *alignment*, 0.25 for *grouping*, 0.1 for *repartition*, 0.1 for *simplicity*, 0.1 for *economy*, 0.005 for *proportion*.

The weights assigned to each measure are an arbitrary estimate of their importance in relation to each other. With *balance* being the most important (according to [23]). The absence of clear and precise amounts for each weight makes the final ranking highly prone to bias and requires further studies to determine what these values should be. Those studies, to the best of our knowledge, do not exist yet. However, considered individually, the metrics do detect the deficiencies they were supposed to detect on the interfaces, and the results do coincide reasonably well with what the users have reported during the interviews:

- The aligned version of the printer GUI is slightly better than the original, but that is barely noticeable.
- The balanced version of Linux GUI is slightly better, but that is barely noticeable.
- Simplicity, Economy, and Grouping rank the Linux GUIs above the Printer dialog boxes as expected and confirmed by the users.
- All metrics ranked Eclipse as the best interface or at least as second best, except for Proportion, Density, and Economy. (and this is justified because of the length of the form fields, and the number of elements and the diversity of their types).

So these metrics are estimated useful^b guidelines for usability testing and should be included in quantitative research. However, future work should improve them to include more advanced aspects that are left out (e.g., typography, color psychology, texture effects and patterns, perspective, and what multimedia content adds to the entire composition). Moreover, future work should combine these measures of aesthetics with usability and utility (or usefulness) in a way that takes into account how they all influence each other in the experience that the user has of a system.

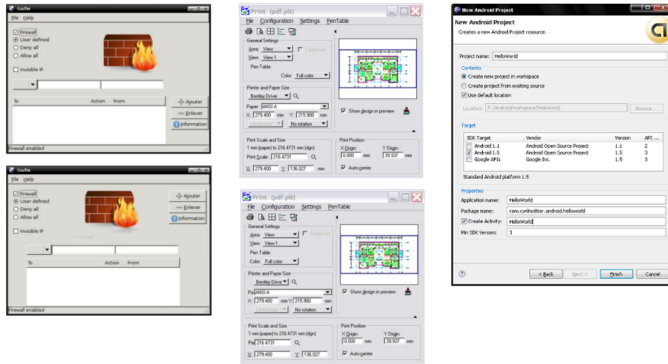


Figure 5: From left to right: Linux GUI and its balanced version, Printer dialog box and its aligned version, an Eclipse UI.

^bAs economist George Box said in a 1976 article, “all models are wrong, but some are useful.” Aesthetic metrics, like any model or scientific theory, do not represent absolute truth but derive their value from their level of usefulness.

6 CONCLUSION

In this paper, we presented AESTHETIXML, a process for computing aesthetics at the level of a Concrete User Interface instead of Final User Interface to make it as much platform independent as possible, in four steps: reverse engineer a final user interface into its concrete equivalent, optionally edit it, and automatically compute ten aesthetic metrics at the level of concrete user interfaces. We provided the mathematical definition of these metrics, as developed in the Restful API. This web service can be reused by another application. Since AESTHETIXML raised the level of abstraction to a CUI, it is located at the syntactical level in Fig. 2.

REFERENCES

- [1] Ahamed Altaboli and Yingzi Lin. 2011. Objective and Subjective Measures of Visual Aesthetics of Website Interface Design: The Two Sides of the Coin. In *Proceedings of 14th International Conference on Human-Computer Interaction. Design and Development Approaches (HCI International '11)*, Julie A. Jacko (Ed.). Springer, Berlin, Heidelberg, 35–44. https://doi.org/10.1007/978-3-642-21602-2_4
- [2] Ahlem Assila, Káthia Marçal de Oliveira, and Houcine Ezzedine. 2016. An environment for integrating subjective and objective usability findings based on measures. In *Proceedings of the IEEE Tenth International Conference on Research Challenges in Information Science (RCIS '16)*, 1–12. <https://doi.org/10.1109/RCIS.2016.7549320>
- [3] Maxim Bakaev, Sebastian Heil, Vladimir Khvorostov, and Martin Gaedke. 2019. Auto-Extraction and Integration of Metrics for Web User Interfaces. *Journal of Web Engineering (JWE)* 17 (03 2019), 561–590. <https://doi.org/10.13052/jwe1540-9589.17676>
- [4] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (6 2003), 289–308. [https://doi.org/10.1016/S0953-5438\(03\)00010-9](https://doi.org/10.1016/S0953-5438(03)00010-9)
- [5] David Chek Ling Ngo and John Byrne. 2001. Another look at a model for evaluating interface aesthetics. *Int. J. Appl. Math. Comput. Sci* 11 (01 2001), 515–535.
- [6] David Chek Ling Ngo, Lian Seng Teo, and John G. Byrne. 2000. A Mathematical Theory of Interface Aesthetics. *Visual Mathematics* 10, 8 (2000), 0–0. <http://eudml.org/doc/256723>
- [7] Sophie Dupuy-Chessa, Yann Laurillau, and Eric Céret. 2016. Considering Aesthetics and Usability Temporalities in a Model Based Development Process. In *Actes de La 28ième Conférence Francophone Sur l'Interaction Homme-Machine (IHM '16)*. Association for Computing Machinery, New York, NY, USA, 25–35. <https://doi.org/10.1145/3004107.3004122>
- [8] Victor López-Jaquero Francisco Montero and Pascual González. 2011. Model-based Reverse Engineering of Legacy Applications User Interfaces. In *Proceedings of the 2nd International Workshop on User Interface Description Languages* (September 6, 2011) (UIDL '11). Thalès, Paris, 128–133.
- [9] Knut Hartmann, Timo Götzelmann, Kamran Ali, and Thomas Strothotte. 2005. Metrics for Functional and Aesthetic Label Layouts. In *Smart Graphics*, Andreas Butz, Brian Fisher, Antonio Krüger, and Patrick Olivier (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 115–126.
- [10] Streveler D. J. and Wasserman A. I. 2014. Quantitative Measures of the Spatial Properties of Screen Designs. In *Proc. of IFIP TC13 Int. Conf. on Human-Computer Interaction (Interact '84)*. North Holland, Amsterdam, The Netherlands, 81–89.
- [11] A. Lecerof and F. Paterno. 1998. Automatic support for usability evaluation. *IEEE Transactions on Software Engineering* 24, 10 (Oct 1998), 863–888. <https://doi.org/10.1109/32.729686>
- [12] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, and Murielle Florins. 2004. USIXML: A User Interface Description Language Supporting Multiple Levels of Independence. In *Proceedings of Workshops in connection with the 4th International Conference on Web Engineering (ICWE '04). Engineering Advanced Web Applications* (28–30 July, 2004) (DIWE '04), Maristella Matera and Sara Comai (Eds.). Rinton Press, 325–338.
- [13] Salvador González López, Francisco Montero Simarro, and Pascual González López. 2013. *BaLOReS: A Framework for Quantitative User Interface Evaluation*. Springer London, London, 127–143. https://doi.org/10.1007/978-1-4471-5445-7_10
- [14] Rohit Mahajan and Ben Shneiderman. 1997. Visual and textual consistency checking tools for graphical user interfaces. *Software Engineering, IEEE Transactions on* 23 (12 1997), 722 – 735. <https://doi.org/10.1109/32.637386>
- [15] Beanjamin Michotte and Jean Vanderdonckt. 2008. GrafiXML, a Multi-target User Interface Builder Based on UsiXML. In *Proceedings of the Fourth International Conference on Automatic and Autonomous Systems (ICAS '08)*, 15–22. <https://doi.org/10.1109/ICAS.2008.29>
- [16] David Ngo, Azman Samsudin, and Rosni Abdullah. 2000. Aesthetic Measures for Assessing Graphic Screens. *J. Inf. Sci. Eng.* 16, 1 (01 2000), 97–116.
- [17] Jakob Nielsen. 1986. A virtual protocol model for computer-human interaction. *International Journal of Man-Machine Studies* 24, 3 (1986), 301 – 312. [https://doi.org/10.1016/S0020-7373\(86\)80028-1](https://doi.org/10.1016/S0020-7373(86)80028-1)
- [18] Gary Perlman. 1987. An Axiomatic Model of Information Presentation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 31 (09 1987). <https://doi.org/10.1177/154193128703101111>
- [19] Andreas Riegler and Clemens Holzmann. 2018. Measuring Visual User Interface Complexity of Mobile Applications With Metrics. *Interacting with Computers* 30, 3 (04 2018), 207–223. <https://doi.org/10.1093/iwc/iwy008> arXiv:https://oup.prod.sis.lan/iwc/article-pdf/30/3/207/24805365/iwy008.pdf
- [20] N Tractinsky, A.S Katz, and D Ikar. 2000. What is beautiful is usable. *Interacting with Computers* 13, 2 (2000), 127 – 145. [https://doi.org/10.1016/S0953-5438\(00\)00031-X](https://doi.org/10.1016/S0953-5438(00)00031-X)
- [21] Thomas S. Tullis. 1983. The Formatting of Alphanumeric Displays: A Review and Analysis. *Human Factors* 25, 6 (1983), 657–682. <https://doi.org/10.1177/001872088302500604>
- [22] Silvia Uribe, Federico Álvarez, and José Manuel Menéndez. 2017. User's Web Page Aesthetics Opinion: A Matter of Low-Level Image Descriptors Based on MPEG-7. *ACM Trans. Web* 11, 1, Article Article 5 (March 2017), 25 pages. <https://doi.org/10.1145/3019595>
- [23] Jean Vanderdonckt and Xavier Gillo. 1994. Visual Techniques for Traditional and Multimedia Layouts. In *Proceedings of the ACM International Workshop on Advanced Visual Interfaces (AVI '94)*. Association for Computing Machinery, New York, NY, USA, 95–104. <https://doi.org/10.1145/192309.192334>
- [24] Mathieu Zen and Jean Vanderdonckt. 2014. Towards an evaluation of graphical user interfaces aesthetics based on metrics. In *Proceedings of the IEEE Eighth International Conference on Research Challenges in Information Science* (May 28–30, 2014) (RCIS '14), 1–12. <https://doi.org/10.1109/RCIS.2014.6861050>
- [25] Óscar Sánchez Ramón, Jean Vanderdonckt, and Jesús García Molina. 2013. Re-engineering graphical user interfaces from their resource files with UsiResourcer. In *Proceedings of the IEEE 7th International Conference on Research Challenges in Information Science* (May 29–31, 2013) (RCIS '13), 1–12. <https://doi.org/10.1109/RCIS.2013.6577696>

APPENDIX. EXAMPLE OF METRICS.

The following table reproduces values of some metrics computed on good and bad user interfaces at the Concrete User Interface (CUI) level, which could differ from those computed

at the Final User Interface (FUI) level. A demonstration example is accessible at <https://www.youtube.com/watch?v=yYNbgwXwOGk>.

Measure (Code)	Value	Bad user interface		Good user interface	
		Interpretation	Value	Interpretation	
Vertical balance (BMv)	0.6348	Low/medium: vertically unbalanced, more weight on the left side	0.0	High: vertically balanced	
Horizontal balance (BMh)	0.6509	Low/medium: horizontally unbalanced, more weight on the top half	0.0074	High: horizontally balanced	
Balance (BM)	0.3571	Low/medium: unbalanced	0.9962	High: balanced	
Vertical equilibrium (EMx)	0.1009	High: equal layout and frame x-coordinates	0.0	High: equal layout and frame x-coordinates	
Horizontal equilibrium (EMy)	0.2936	High: equal layout and frame y-coordinates	0.0	High: equal layout and frame y-coordinates	
Equilibrium (EM)	0.8027	High: stable	1.0	High: stable	
Vertical symmetry (SYMv)	0.6456	Low/medium: vertically asymmetrical	0.0	High: vertically symmetrical	
Horizontal symmetry (SYMh)	0.6274	Low/medium: horizontally asymmetrical	0.0022	High: horizontally symmetrical	
Radial symmetry (SYMr)	0.3725	Medium/high: radially symmetrical	0.0022	High: radially symmetrical	
Symmetry (SYM)	0.4514	Medium: asymmetrical	0.9984	High: symmetrical	
Sequence (SQM)	0.5	Medium: random	1.0	High: sequential	
Cohesion of objects (CMlo)	0.3882	Low/medium: unequal object and layout aspect ratios	0.7166	High: equal object and layout aspect ratios	
Cohesion of frames (CMfl)	0.9704	High: equal layout and frame aspect ratio	0.8856	High: equal layout and frame aspect ratios	
Cohesion (CM)	0.6793	Medium/high: cohesive	0.8011	High: cohesive	
Unity of forms (UMform)	0.1428	Low: many object sizes	0.9	High: few object sizes	
Unity of space (UMspace)	0.0728	Low: less space left at the margins	0.8533	High: less space between objects	
Unity (UM)	0.1078	Low: fragmented	0.8766	High: unified	
Proportion of objects (PMo)	0.6001	Medium: aesthetically pleasing objects	0.8231	High: aesthetically pleasing objects	
Proportion of layout (PML)	0.8686	High: aesthetically pleasant layout	0.9140	High: aesthetically pleasing layout	
Proportion (PM)	0.7344	High: proportionate	0.8685	High: proportionate	
Complexity (SMM)	0.1428	Low: complex	0.1764	Low: complex	
Space (DM)	0.4153	Medium: cramped	0.8218	High: spacious	
Regularity of alignment (RMa)	0.0	Low: many alignment points	0.65	Medium/high: few alignment points	
Regularity of distances (RMs)	0.1667	Low: many distinct distances between alignment points	0.944	High: few distinct distances between alignment points	
Regularity (RM)	0.0833	Low: irregular	0.7972	High: regular	
Intricacy (ECM)	0.1428	Low: intricate	0.5	Medium: intricate	
(W)	1.1 ^{18}		2.31 ^{12}		
(WMax)	1.93 ^{23}		2.31 ^{12}		
Homogeneity (HM)	5.68 ^{-06}	Low: uneven	1	High: homogeneous	
Rhythm horizontal (RHMx)	0.5855	Medium: rhythmic horizontal arrangement variation	0.0	High: rhythmic horizontal arrangement variation	
Rhythm vertical (RHMy)	0.5531	Medium: rhythmic vertical arrangement variation	0.0047	High: rhythmic vertical arrangement variation	
Rhythm (RHMa)	0.5021	Medium: disorganised dimension variation	0.0	High: disorganised dimension variation	
Rhythm (RHM)	0.4530	Medium: disorganised	0.9984	High: rhythmic	
Overall score (OM)	0.3746	Bad	0.8335	Good	

Recognizing a Sketched Task Model by Multi-Stroke Gesture Recognition

Jorge-Luis Pérez-Medina, Karina-Beatriz Jimenes-Vargas

Universidad de Las Américas (UDLA), Intelligent and Interactive Systems Lab (SI^2 Lab), Quito, Ecuador

{jorge.perez.medina,karina.jimenes}@udla.edu.ec

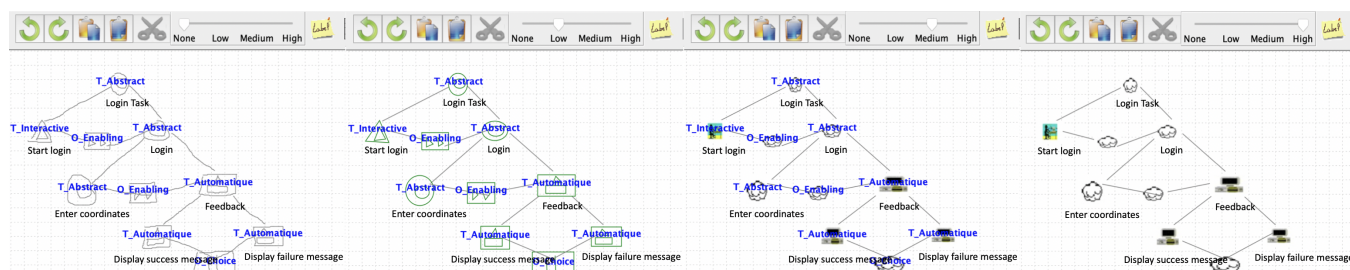


Figure 1: Sketching a task model: (1) draft, (2) after stroke recognition, (3) after beautification, (4) after transformation.

ABSTRACT

TASKSKETCH is an eclipse-based task model editor to enable designers to sketch a task model and to recognize it by a multi-stroke gesture recognition of pre-defined shapes and relations. The tool supports three levels of definition (LoF, LoD, LoC) and is linked to other models by model mapping. TASKSKETCH is flexible enough to accommodate variations of the task model notation by defining new shapes and relations, and by searching template and pattern matchings. To demonstrate this capability, we introduce three concepts: the level of fidelity, the level of detail, and the level of criticism. This process is generalizable to other similar models involved in the model-based user interface design.

CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces**; **User interface toolkits**; *Gestural input*;

KEYWORDS

Sketching; Gesture Recognition; Task Modeling; Design Tools and Techniques

ACM Reference Format:

Jorge-Luis Pérez-Medina, Karina-Beatriz Jimenes-Vargas. 2020. Recognizing a Sketched Task Model by Multi-Stroke Gesture Recognition. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/12345.6789>

1 INTRODUCTION

Model-driven engineering (MDE) approaches have allowed the emergence of trends in automating modeling platforms and thus facilitate the development of many steps in software development [6]. Human-computer interaction (HCI) domain need of MDE [21] approaches to support the design process. MDE supports the early stages of a development process, where sketching facilitates the design of user interfaces (UIs) of an interactive system.

In this context, graphic modeling tools become an essential aspect in the development process. However, the dynamics of the current processes require an agile, and a flexible modeling [2]. Sketching is also particularly challenging when prototyping multi-platform user interfaces, for different contexts of its use [7]. The Cameleon Reference Framework (CRF) [4] specifies that the User Tasks (UT) must be modeled before considering the abstract user interface, the concrete user interface and the final User Interfaces (FUI) of any system.

Nowadays, different tools [3, 5, 11, 29, 30] offer the possibility of prototyping the system user interfaces through sketches. However, this is not the case for tools that allow creating task models. These models are generally crafted using classic tools where the user has to interact with graphical representations through a toolbar. We advocate offering lighter and more flexible tools to facilitate a quick sketches modeling in collaborative work environments, based on diverse spaces [22], such as a wall screen or a tablet.

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-6745-5/20/06...\$15.00

<https://doi.org/10.1145/12345.6789>

To address this need, we propose to the Engineering Interactive Computing Systems (EICS) community the tool **TASKSKETCH** for sketching a task model based on a reusable multi-stroke gesture recognizer exploiting multiple templates based on tasks notation. This tool has the following characteristics:

- A multi-stroke gesture recognition algorithm which supports large surfaces, based on [20].
- Multiple levels of fidelity to visualize the task models elements as shown in Figure 1. The tool supports three levels of definition (LoD, LoC, LoI) and that it is linked to other models by model mapping.
- The option of defining XML grammars to specify the elements of the task models considering a specific notation, for example: ConcurTaskTrees (CTT) [17] (see the external link: <https://github.com/jperezmedina/TaskSketch>). Users can create other models with different combinations of primary shapes by creating a XML-based grammar file.

2 RELATED WORK

Software for Task Modelling

The modeling of end-users' tasks is a well known practice and a widely used in the HCI domain. Academics and industrialists have developed modeling languages such as: MAD [26], CTT [17] and HAMSTERS [1]. Likewise, we find tools such as: Mobi-D [24], CTTe [18], HAMSTERS^a [14] and Hamster-XLE [15]. They allow designers to carry out in an assisted manner the design of task trees.

Overview of Recognizers for Modeling Artefacts

Even though Software Engineering offers already some software to sketch UML diagrams incorporating a multi-stroke sketch recognition algorithm, such as Tahuti [9], the Human-Computer Interaction field does not have good sketching tools relying on gesture recognizers. The current tools, in terms of sketching tools, maintain a low level of fidelity [8].

Paternò [19] presents an approach to support designers to intuitively model interactive applications by converting sketches, hand-drawn sketches, into task specifications that are more editable and parsable. The developed environment allows drawing a task model on a board. Then, the Mimio device captures the image and the environment to transform it into an editable model. This approach forces the designer to draw everything to recognize the drawing.

Other types of tools enable the quick prototyping activities of sketch-based user interfaces. In this context, The best-known tools are: JavaSketchIt [3], sketchiXML [5], FlexiSketch [29], and FlexiSketch TEAM [30].

By contrast, **TASKSKETCH** is not limited to the use of one model. The tool is capable to keep part of the drawing in

low fidelity [23]. **TASKSKETCH** performs the recognition of gestures in real time and allows the users to move from one fidelity level to another, or mix levels. The tool supports three levels of definition (LoF, LoD, LoC) and that it is linked to other models by model mapping [16].

The environment of **TASKSKETCH** is flexible, so we can draw different levels of fidelity, details and criticality based on conceptual representations. It supports the following fidelity levels: (1) drawing fidelity: the designs are displayed as were drawn, nothing else; (2) shape fidelity: recognized primary shapes (or their combinations) are smoothed, modeling elements compliant with a specification are not displayed; (3) modeling fidelity: modeling elements are displayed as representative images, the remaining forms as smooth representations; (4) high fidelity: modeling elements are displayed as real task modeling elements, the remaining forms in smoothed representation.

3 EXAMPLES OF REPRESENTATIONS

In MDE, a transformation from a CIM model to a PIM model could be ensured through animated transitions and similarly from a PIM model to a PSM model, and from a PSM model to code. Similarly in CRF, animated transitions could be used from task and concepts to abstract UI, from abstract UI to concrete UI, and from concrete UI to final UI. Since the development life cycle should not necessarily go through all these levels, animated transitions could be created between non-subsequent levels of abstraction, such as from CIM to PSM, PIM to code, or CIM to code. In CRF, it could be from task and concepts to concrete UI, from abstract UI directly to final UI. Depending on the type of model attribute, on the type of corresponding shape (e.g., a rectangle, a triangle, a circle, a line), different animated transitions could be imagined such as: text-to-text, text-to-position, text-to-dimension, text-to-color, text-to-shape or reciprocal transitions [27].

The fidelity

Figure 2 reproduces a conceptual representation (a task model as defined in CRF) with different levels of fidelity, details, and criticality. The cell #3 depicts a simple task model in its final format (high fidelity) where a task is decomposed into two sub-tasks: (1) an interactive one (depicted by an interactivity icon) and (2) a computer one (depicted by a computer icon) with a sequence operator between (»). Anyone can sketch a model (cell #1 - low fidelity), it is recognized by the software and beautified (cell #2 - medium fidelity), and then rendered into its final format (cell #3 - high fidelity).

The details

When a person sketch a task model at an early design, there is no need to have the final format: an informal format is sufficient. The design process evolves towards more advanced

^aHuman - centered Assessment and Modelling to Support Task Engineering for Resilient Systems

	Low	Medium	High
Fidelity	1 raw sketches 	2 beautification 	3 rendered 
Details	4 no recognition 	5 lazy recognition 	6 eager recognition 
Criticality	7 	8 	9 

Figure 2: A conceptual representation with different levels of fidelity, details and Criticality.

and detailed aspects, however, this format should become more formal. Cells #4, #5, #6 represents the same task model with different levels of details: cell #4 provides the initial sketch that is naturally drawn, cell #5 uses lazy recognition to render properties by line style and color (i.e., green), and cell #6 uses eager recognition for the final format.

The criticality

Other model elements of the task model could be rendered equally: cells #7, #8, and #9 depicts the level of criticality (a particular case of LoD). The thicker the line width is, the more critical the task is; the first sub-task is more critical than the second one.

During the UI development life cycle, a group of task analysts may start with representations #1, #4, and #7, sketching during a meeting until, later on the development, they decide the model has an acceptable LoD, moving towards its integration with other models. They can raise the LoD level to medium, and the different elements of the task model are recognized and could be linked with other models. UI designers could start with representations #3, #4, or #7, with an already done UI on a high LoD, and lower the LoD to discuss with users on a more abstract level. The users can sketch their own ideas on a low LoF, which can be transposed later to higher LoF. However, to obtain a true co-evolutionary design we must consider that UI designers and other stakeholders will not be working on the same representations at the same LoF and LoD. Therefore, the possible approaches for multiple models are detailed bellow, from the simplest case (one model, with single level of fidelity/details/criticality, and producing a single, eventually different, level in the output) to the actual state, where people work at many models on different levels [28].

4 THE TASKSKETCH TOOL

TASKSKETCH is an eclipse plugin that runs on cross-platform which supports sketching recognition on different surfaces. It offers the ability to recognize predefined shapes (i.e., line, arrow, cross, wavy line, triangle, rectangle, ellipse, and circle)

and to combine shapes and has available six basic commands to support the modeling edition (i.e., undo, redo, copy, paste, cut, new window). Also, it supports the visualization of multiple fidelity modeling levels (i.e. none, low, medium and high) and multiple trait recognition.

The tool is configured to recognize ConcurTaskTrees and HAMSTERS, however, a designer can define another series of grammar. The high fidelity elements are the graphical representation of the tasks notations (i.e., ConcurTaskTrees). The element recognition is performed when the user drawing on a canvas through a multi-stroke gesture recognizer algorithm which is based on supervised learning.

Software Specifications

TASKSKETCH is based on a natural User eXperience (UX) and takes account that a lower fidelity level is more pleasant during the drawing. The designers can draw the five geometric shapes, such as arrow, line, circle, rectangle and triangle with several strokes. The tool recognizes the geometric shapes and transform them into a smoothed shape. A smoothed representation has a higher fidelity level that its equivalent freehand-drawn and is represented by an image. Also, it is able to combine multiple simple shapes in a more complex way according to the pre-established grammars. If all the constraints of a specific grammar are respected, the representation is recognized.

According to the defined grammars, the possible combinations must be mutually exclusives. If two mutually exclusive combinations are possible based on a list of representations, the most constrained combination should be selected. After that, each drawing, TASKSKETCH is analyzed to return the recognized vector shape. The designer may also give a visual feedback, via annotations. Annotations are the only way to write textual phrases during design. Actually, the tool does not perform any recognition when the designer uses the annotation mode. However, when an element is recognized, a pop-up window allows the designer to associate a text.

TASKSKETCH allows to define the modeling elements, so a designer can add or change a composition rule using an XML file. This mechanism facilitate the customization of the modeling elements according to the needs of the designers. During task modeling design, once a designer has created a modeling element, the tool allows him to insert a textual description through text recognition.

TASKSKETCH is also consistent with the current model drawing, it means that, the tool is compliant with “what you see is what you get” (WYSIWYG). The tool offers the possibility to select a modeling element (or groups of elements) in case of require aligning or duplicate them.

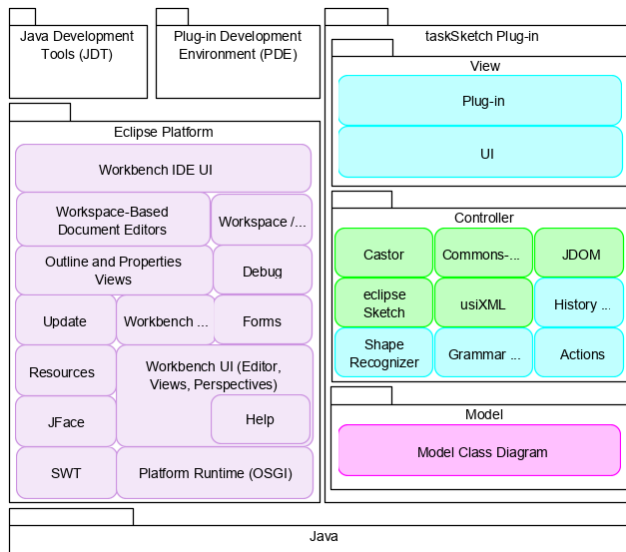


Figure 3: Software architecture of TASKSKETCH.

The TASKSKETCH software architecture

Figure 3 shows the global architecture of the TASKSKETCH plugin which is based on the Eclipse architecture. The plugin *View Package* was created considering the MVC model and contains the UIs standards implemented with the Eclipse SDK. The *Plug-in Package* allows TASKSKETCH to connect with the elements of the SDK Eclipse platform. The *UI Package* manages the perspective of the tool, that is, all the application views. The main views are: the menu and toolbar, the windows edition view and the perspective view.

The *Controller Package* connects with the View and Model packages. It contains the functional elements developed and the external libraries integrated to the plugin.

The *Shape Recognizer package* manages the intelligent thread which recognizes the best conditions for the pre-defined shapes or the combinations of them. When a condition is obtained, it is converted into a smoothed representation. The *Shape Recognizer package* also recognizes a form based on a set of points implemented by the class *DotSet* contained into the model. The *Grammar Recognizer package* is in charge of managing the structures of the used pre-established rules to define the modeling elements.

Each grammar is defined as follows: each form used is a variable whose initial domain is the set of the representations contained in the canvas being analyzed. These areas are reduced progressively until they contain only a member; the variable is then considered to be linked. Once all the variables (i.e. all the representations involved in the grammar) are linked, the grammar is tested. If the grammar is valid, the solution is stored in a list of acceptable solutions. The *Grammar Recognizer package* then continues searching in linking variables, otherwise, until all possible solutions have been tested. The constraint's evaluation is based on

[10] due to its ability to prune the search tree by constraint propagation, which means that, every constraint is treated independently, and if certain combination do not respect a restriction, they may be removed.

The *Action and the History Manager packages* manages the events carried out by the UI elements contained into the *View Package*.

TASKSKETCH re-uses several external libraries. The external libraries are listed below. The Open Source *Castor*^b data binding framework allows using XML into Java applications. The Apache Commons Logging (JCL)^c provides a Log of all events on the tool. The shape recognition is based on the *Eclipse Sketch* shape recognition developed by [25]. We have adapted this library to recognize shapes or combinations of them on large surfaces. *UsiXML* allows you to analyze and export forms to the Extensible Markup Language of User Interface (usiXML) [12]. JDOM^d is required by UsiXML and provides support to manipulate XML data from Java code.

Finally, The *Model Package* is represented by the Model Class Diagram which is described in the next section.

The Model Class Diagram

Figure 4 partially reproduces the Model Class Diagram contained into the Model Package of TASKSKETCH. The *Graphics* class is the superclass of the tool structures required for storage the displayed representations. *Graphics* class is specialized by the classes *Vectorial Shape* and *DotsSet*, *Graphics Container* and *Widget*.

The *Graphics Container* Class is the structure to manage the *Windows* elements and the *Shape Events*. The *Vectorial Shape* class represents the 6 pre-defined basic shapes (i.e., arrow, line, wavy line, ellipse, triangle, rectangle). This class allows manipulating the geometric shapes and converting them in smoothed representations. The *Widget* class is the constitutive element of a Graphical User Interface (GUI). It class have associated an Image. Several classes inherit from *Widget*. It can be a button, a text, a text field, a listBox, etc. The purpose of this class is to inform the user, or allow him to interact with the system used. Widgets are recon

The *Task Element* class represents the structure of the elements defined by the different notations (i.e., ConcurTask-Trees, HAMSTERS and/or others). We have left the possibility of being able to specialize this class in order to extend the grammar and to incorporate new modeling languages (i.e., UML, Widgets, ...). Each element inherited from the Task Element class is associated with a reference image in order to offer a better high fidelity display. The *DotsSet* class elements represent a hand drawing performed by the user.

^bSee <http://castor-data-binding.github.io/castor/>

^cSee <https://commons.apache.org/proper/commons-logging/>

^dSee <http://www.jdom.org/>

This element is a set of sequentially connected points and is used as annotations. Widgets are recognized by the *Shape Recognizer* class by means of a distance. We currently use Levenstein distance.

The *Annotations Container* class is the structure of storage for elements of type *DotsSet*. The *Windows* class represents an element associated with the *GraphicsContainer*, the *AnnotationContainer* and the *HistoryManager*.

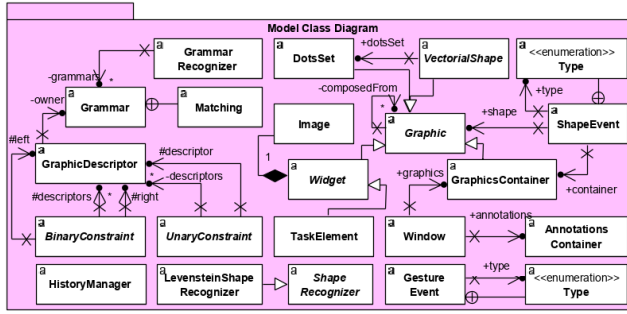


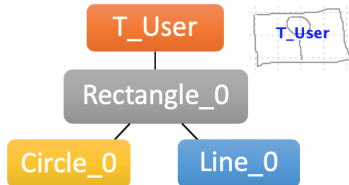
Figure 4: The Graphics Class Diagram of TASKSKETCH.

```
<representation id="T_User">
  <shape id="Rectangle_0" type="Rectangle" />
  <shape id="Circle_0" type="Circle, Ellipse" />
  <shape id="Line_0" type="Line" />
  <binaryConstraint id="0" shape1="Circle_0"
    condition="IsInside" shape2="Rectangle_0"
    threshold="0.9" />
  <binaryConstraint id="1" shape1="Line_0"
    condition="IsInside" shape2="Rectangle_0"
    threshold="0.9"/>
  <unaryConstraint id="2" shape="Line_0"
    condition="isVertical" />
  <binaryConstraint id="5" shape1="Line_0"
    condition="isOnTheBottomOf" shape2="Circle_0"
    threshold="0.9" />
</representation>
```

(a) The XML grammar for the User's Task.

```
<?xml version="1.0" encoding="UTF-8"
  standalone="no"?>
<!DOCTYPE properties SYSTEM
  "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="55533331117777">Rectangle</entry>
  <entry key="234356712">Circle</entry>
  <entry key="5555">Line</entry>
</properties>
```

(b) The gestures representation the User's Task.



(c) The graphical data structure of the User's Task.

Figure 5: The User's Task representation.

A specialized XML file allows defining a set of rules for the graphics elements, except the *DotsSets*. Figure 5a presents an example of a structure that contains the definition of a User Task representation. The three basic predefined shapes used to define a user's task are: a rectangle, a circle and a line. The relationship between these elements is denoted by three

binary constraints. Binary constraint tests if a specific relation between two predefined shapes is verified. For instance, the first binary *IsInside* constraint of the XML grammar evaluates if the *Circle_0* is inside the *Rectangle_0*. The unitary constraints apply only to a particular element at a time. For instance: the unitary constraint *isVertical* tests if the *Line_0* is considered vertical.

Figure 5b shows an XML file containing the three predefined shapes used in the recognition of the user's Task. These representations are based on wind rose described by [13]. Figure 5c shows a graphical idea of the same user's task data structure. It corresponds to an inverted tree whose nodes are graphic elements. In particular the leaves are *Vectorial Shapes* instances recognized by the multiple stroke recognizer algorithm. The final representation of recognized drawn shapes is always performed in the leaves. Finally, when rendering, the path of this structure behaves differently depending on the level of fidelity, according to the levels (see Figure 6).

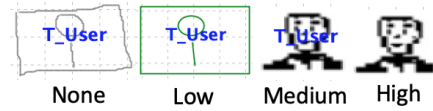


Figure 6: Possible levels of fidelity for a User's Task.

Figure 1 presents an example corresponding to the functionality of Login. We use this functionality to demonstrate the applicability of the TASKSKETCH tool. While the sketches are made, the tool recognizes all possible combinations of possible. Optionally, designers can insert a task description when it is recognized. To perform the recognition of user gestures, the tool uses a grammar that describes each of the task model elements according to the CTT notation [17]. The designer can navigate between the different levels of fidelity in order to obtain a visual representation more refined and adjusted to the notation CTT.

5 CONCLUSION

We present the technical aspects of TASKSKETCH. This Eclipse plugin is different from other sketching tools in that it supports three levels of definition (LoF, LoD, LoC) and that it is linked to other models by model mapping. Also, its use is not limited by one model. Users can create as many models as they like from different combinations of primary shapes into a XML-based grammar file. Gestures represented by multi-stroke gestures are recognized and transformed into elements of the task model according to a user-defined grammar into an XML format.

With the use of this tool, we advocate a more natural, flexible, customizable, agile and, an adaptable mechanism to new collaborative design approaches in which multiple design spaces [22] can be used. The screen size, of the device configurations, supported by TASKSKETCH can range from

small to wall screen. TASKSKETCH's XML grammar can be augmented by representations that allow recognizing the components of a UI. In this sense, the tool could support prototyping activities, thus, designers and stakeholders can model, in parallel, the tasks and its associated user interface. We conclude by indicating that the presented tool can certainly serve as the foundation for considerable expansion and future engineering and experimental work involving designers and end-users.

REFERENCES

- [1] Eric Barboni, Jean-François Ladry, David Navarre, Philippe Palanque, and Marco Winckler. 2010. Beyond modelling: an integrated environment supporting co-execution of tasks and systems models. In *Proc. of the 2nd ACM SIGCHI symposium on EICS*. 165–174.
- [2] Matthias Book and André van der Hoek. 2018. Sketching with a purpose: Moving from supporting modeling to supporting software engineering activities. In *2018 IEEE/ACM 11th Int. Workshop on Co-operative and Human Aspects of Software Engineering (CHASE)*. IEEE, 93–96.
- [3] Anabela Caetano, Neri Goulart, Manuel Fonseca, and Joaquim Jorge. 2002. Javasketchit: Issues in sketching the look of user interfaces. In *AAAI Spring Symposium on Sketch Understanding*. 9–14.
- [4] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (6 2003), 289–308.
- [5] Adrien Coyette, Stéphane Faulkner, Manuel Kolp, Quentin Limbourg, and Jean Vanderdonckt. 2004. SketchiXML: towards a multi-agent design tool for sketching user interfaces based on USIXML. In *Proc. of the 3rd annual conference on Task models and diagrams*. 75–82.
- [6] J Eisenstein, J Vanderdonckt, and A Puerta. 2001. Model-based user-interface development techniques for mobile computing. In *Proc. of 5th ACM Int. Conf. on Intelligent User Interfaces IUI*, Vol. 1. 69–76.
- [7] Murielle Florins, Francisco Montero Simarro, Jean Vanderdonckt, and Benjamin Michotte. 2006. Splitting rules for graceful degradation of user interfaces. In *Proc. of the working conference on Advanced visual interfaces*. 59–66.
- [8] Jean-Luc Hak, Marco Winckler, and David Navarre. 2016. PANDA: Prototyping Using Annotation and Decision Analysis. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '16)*. Association for Computing Machinery, New York, NY, USA, 171–176. <https://doi.org/10.1145/2933242.2935873>
- [9] Tracy Hammond and Randall Davis. 2006. Tahuti: A geometrical sketch recognition system for uml class diagrams. In *ACM SIGGRAPH 2006 Courses*. 25–es.
- [10] Robert M Haralick and Gordon L Elliott. 1980. Increasing tree search efficiency for constraint satisfaction problems. *Artificial intelligence* 14, 3 (1980), 263–313.
- [11] James A Landay and Brad A Myers. 2001. Sketching interfaces: Toward more human interface design. *Computer* 34, 3 (2001), 56–64.
- [12] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, and Murielle Florins. 2004. USIXML: A User Interface Description Language Supporting Multiple Levels of Independence. In *Proc. of Workshops in connection with the 4th Int. Conf. on Web Engineering (ICWE '04)* (28-30 July, 2004) (*DIWE '04*), Maristella Matera and Sara Comai (Eds.). Rinton Press, 325–338.
- [13] David Llorens, Federico Prat, Andrés Marzal, Juan Miguel Vilar, María José Castro, Juan-Carlos Amengual, Sergio Barrachina, Antonio Castellanos, Salvador Espana Boquera, JA Gómez, et al. 2008. The UJlpenchars Database: a Pen-Based Database of Isolated Handwritten Characters.. In *LREC*.
- [14] C Martinie, P Palanque, E Barboni, M Winckler, M Ragosta, and Paola Lanzi Alberto Pasquini. 2011. Formal tasks and systems models as a tool for specifying and assessing automation designs (regular paper). In *1st Int. Conf. on Application and Theory of Automation in Command and Control Systems (ATACCS 2011) Barcelona, Spain*.
- [15] Célia Martinie, Philippe Palanque, Elodie Bouzekri, Andy Cockburn, Alexandre Canny, and Eric Barboni. 2019. Analysing and demonstrating tool-supported customizable task notations. *Proceedings of the ACM on Human-Computer Interaction* 3, EICS (2019), 1–26.
- [16] Francisco Montero, Víctor López-Jaquero, Jean Vanderdonckt, Pascual González, María Lozano, and Quentin Limbourg. 2005. Solving the Mapping Problem in User Interface Design by Seamless Integration in I dealXML. In *Int. Workshop on Design, Specification, and Verification of Interactive Systems*. Springer, 161–172.
- [17] Fabio Paternò, Cristiano Mancini, and Silvia Meniconi. 1997. Concur-TaskTrees: A diagrammatic notation for specifying task models. In *Human-computer interaction INTERACT'97*. Springer, 362–369.
- [18] Fabio Paternò, Giulio Mori, and Riccardo Galiberti. 2001. CTTE: an environment for analysis and development of task models of cooperative applications. In *CHI'01 Extended Abstracts on Human Factors in CS*. 21–22.
- [19] Fabio Paternò and Marco Volpe. 2005. Natural modelling of interactive applications. In *Int. Workshop on Design, Specification, and Verification of Interactive Systems*. Springer, 67–77.
- [20] Jorge-Luis Pérez-Medina. 2016. Methods for Modelling Sketches in the Collaborative Prototyping of User Interfaces. *Revista Romana de Interactiune Om-Calculator* 9, 3 (2016), 183–216.
- [21] Jorge-Luis Pérez-Medina, Sophie Dupuy-Chessa, et al. 2007. A survey of model driven engineering tools for user interface design. In *Int. Workshop on Task Models and Diagrams for User Interface Design*. Springer, 84–97.
- [22] Jorge-Luis Pérez-Medina, Jean Vanderdonckt, and Santiago Villarreal-Narvaez. 2019. CROSSIDE: A Design Space for Characterizing Cross-Surface Collaboration by Sketching. *Journal of Visual Language and Computing* 1 (2019), 29–41.
- [23] Jennifer N Petrie and Kevin A Schneider. 2006. Mixed-fidelity prototyping of user interfaces. In *Int. Workshop on Design, Specification, and Verification of Interactive Systems*. Springer, 199–212.
- [24] Angel Puerta and Jacob Eisenstein. 1999. Towards a general computational framework for model-based interface development systems. *Knowledge-Based Systems* 12, 8 (1999), 433–442.
- [25] Ugo Braga Sangiorgi and SD Barbosa. 2010. SKETCH: modeling using freehand drawing in eclipse graphical editors. *FlexiTools@ ICSE* (2010).
- [26] D Scapin and C Pierret-Golbreich. 1990. Towards a method for task description: MAD, Work with display units 89, L. Berlinguet, D. Berthelett Eds.
- [27] J. Vanderdonckt. 2012. Animated transitions for empowering interactive information systems. In *2012 Sixth Int. Conf. on Research Challenges in Information Science (RCIS)*. 1–12.
- [28] Jean Vanderdonckt and Pierre Berquin. 1999. Towards a Very Large Model-Based Approach for User Interface Development. In *Proc. of the 1999 User Interfaces to Data Intensive Systems (UIDIS '99)*. IEEE Computer Society, USA, 76.
- [29] Dustin Wüest, Norbert Seyff, and Martin Glinz. 2012. Flexisketch: A mobile sketching tool for software modeling. In *Int. Conf. on Mobile Computing, Applications, and Services*. Springer, 225–244.
- [30] Dustin Wüest, Norbert Seyff, and Martin Glinz. 2015. Flexisketch team: Collaborative sketching and notation creation on the fly. In *2015 IEEE/ACM 37th IEEE Int. Conf. on Soft. Eng.*, Vol. 2. IEEE, 685–688.

A Pen User Interface for Controlling a Virtual Puppet

Jean Vanderdonckt

Université catholique de Louvain (UCLouvain), LouRIM
Louvain-la-Neuve, Belgium
jean.vanderdonckt@uclouvain.be

Radu-Daniel Vatavu

MintViz Lab, University Ștefan cel Mare of Suceava
Suceava, Romania
radu.vatavu@usm.ro

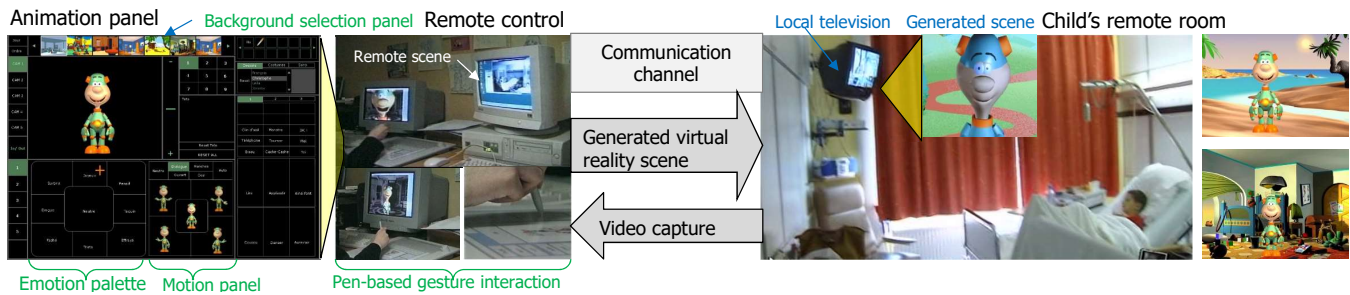


Figure 1: Context of use for our application: a psychologist, from a remote location, controls a virtual puppet via an animation panel and pen gestures (left) generating a Virtual Reality story displayed in the hospital room of a child (right).

ABSTRACT

We present an application and corresponding user interface where the user, a psychologist, controls a virtual puppet (a cartoon-like character in Virtual Reality) that displays human-like emotions for the benefit of hospitalized children. A crossing-based UI enables specification of emotional states for the virtual puppet, which are chosen from a widget UI control showing several categories of body poses, movements, and emotions. Input is performed with pen gestures that trigger animated transitions between the states of the virtual puppet. Gesture articulation parameters, such as gesture speed and the pressure applied on the touchscreen, control the rendering of the various animations. We discuss a practical application for the pediatric departments of hospitals.

CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces; Gestural input; Pointing device; Touch screens**; • **Software and its engineering** → Virtual worlds software;

Permission to make digital or hard copies of all or part 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 permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398637>

KEYWORDS

Pen gesture input, Virtual puppetry, Digital puppetry, Virtual Reality, Remote manipulation, User interface.

ACM Reference Format:

Jean Vanderdonckt and Radu-Daniel Vatavu. 2020. A Pen User Interface for Controlling a Virtual Puppet. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398637>

1 INTRODUCTION

Virtual Reality (VR) is expected to deliver a sense of deep presence by displaying photorealistic scenes and characters, such as virtual puppets [6, 12, 14] that exhibit human-like behavior [16]. It is known that humans tend to anthropomorphize virtual characters by projecting their own emotional states [18]. Although simulations of human-like emotions can be rendered in ways that look realistic, there is always the risk of psychological rejection when users notice discrepancies with their own expectations of human-like interactions, a phenomenon known as the “uncanny valley” [15]. Therefore, the behavioral design of virtual characters should compromise between displaying human-like emotions at a level that is sufficient enough to trigger emotional responses and life-like behavior in a ultra-realistic virtual scene, which may induce a negative response from the user [17]. In this paper, we propose an application and user interface in which the user, a psychologist, searches for the optimum balance between the two when manipulating a cartoon-like character for the benefit of hospitalized children; see Fig. 1.

2 RELATED WORK

Since cartoon-like characters that display emotions trigger more responses in humans than those without an emotional/behavioral component, it is important to convey emotion effectively. From the wide literature available on this topic, we identified two references useful to us. The first one, the Hierarchical Model of Affect (HMA) [10], decomposes human emotions into three layers (see Fig. 2a):

- (1) *Emotions* are short-term human affects that last for a short time, after which they vanish from perception.
- (2) *Moods* are negative or positive medium-term affects that are conveyed by the underlying emotions.
- (3) *Personality* is a long-term affect produced by the dominant moods over a longer time period.

This model is useful to understand how to assign moods or a personality to a virtual character, but does not specify which of the emotions are basic affects. To address the latter aspect, we use the Arousal-Valence model [18] that projects emotions on two dimensions (Fig. 2b): arousal shows how exciting an emotion is and valence how positive it is.

Gesture-based User Interfaces

Since our virtual character is controlled by means of gesture input, we overview in the following aspects regarding gesture-based interaction. A *flick* is represented by a quick stroke in a specified direction: *e.g.*, a left flick to delete an item from a list, flicks up and down to scroll a list of items, and diagonal flicks are used to indicate targets [5, 7]. When successive flicks are combined, they form unistrokes, such as the gestures used in *marking menus* [26]. Zhao & Chignell [26] demonstrated how a suite of flicks can form a marking menu, *e.g.*, a Chinese style glyph, which positively influenced user performance for item selection tasks. The visual appearance of marking menus assists with their navigation, while the grouping and categorization of menu items according to their visual salience increases remembrance. In most applications involving *stroke-gesture input*, gesture templates are employed for classification purposes [5, 7, 21, 25, 25]. However, the same gesture type can be articulated in various ways in terms of its location inside the active sensing area, its size, orientation, and so on [2, 20]. Independence of gesture recognizers with respect to such articulation details offers great input flexibility for end users [19, 20]. Integrating pen-gestures in graphical user interfaces can be achieved in different ways, such as by package, toolkit, or API [4].

Digital Puppetry

Gestures have been considered for manipulating virtual puppets. For example, EUCLIDE [6] exploited a combination of multiple devices (data glove, MIDI music board, keyboard, and mouse) for the multimodal control of a puppet in VR.

Barrientos [3] developed an application to transfer gestural attitudes to an avatar; the avatar behaved like a virtual puppet manipulated by strings and the system tracked the motion gestures of the user and projected them onto the avatar. Oshita [17] developed a human-like virtual character with expressions controlled by the pressure and tilt specified with an interactive pen: the character could walk, run, and turn along a specified trajectory according to the speed of the movement induced on the pen.

The Virtual Puppetry system [14] deployed computer-assisted storytelling systems by exploiting motion sensing and gesture control to enable multiple users to narrate and interact with virtual characters in the VR scene. The Virtual Marionette [12] was a study on digital puppetry taking the form of an interdisciplinary approach that brought together the art of puppetry and digital animation. In the Mani-Pull-Action system [13], Leap Motion was used to acquire 3-D gestures performed with the hand to control a cinematic interactive application inspired by the art of marionettes [24]. This setup mimics the genuine gestures performed by an artist manipulating a real puppet, leaving the hands and fingers free over the Leap Motion, thus representing probably the best match between the reality and the virtual environment for such an artist. Without entering into this level of sophistication, other devices, such as ring devices [9], could be explored to add the third dimension.

3 CONTEXT OF USE OF OUR APPLICATION FOR HOSPITALIZED CHILDREN

The context of use envisioned in this paper for our application concerns the pediatrics departments of hospitals, where children undergo operations that may cause trauma. *Our hypothesis is that children will freely and frankly talk to a virtual character in order to share with it their pains, concerns, and worries* [11]. From a remote location (see Fig. 1), a psychologist initiates the discussion with the child with a voice that is altered (to resemble the voice that the virtual character would have) and operates the virtual character (taking in our case the form of a kind extraterrestrial being, chosen to avoid too much anthropomorphism) displayed on the TV set available in the child's hospital room [11]. Equipped with a headset, a graphic tablet, and a stylus (Fig. 1, left), the psychologist-animator can see the child's reactions at all times through a live video feed. The conversation is implemented via animations of the virtual character (whose lip movements are synchronized with the voice of the animator), a microphone, and an animation widget control panel from the user interface available to the psychologist-animator. The latter permits a wide range of expressions, attitudes, and emotions for the virtual character to display [11].

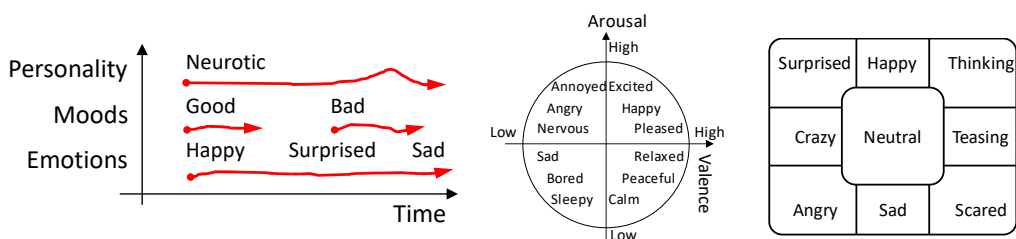


Figure 2: The Hierarchical Model of Affect over time [10], the Arousal-Valence plane [18], and the resulting emotion palette.

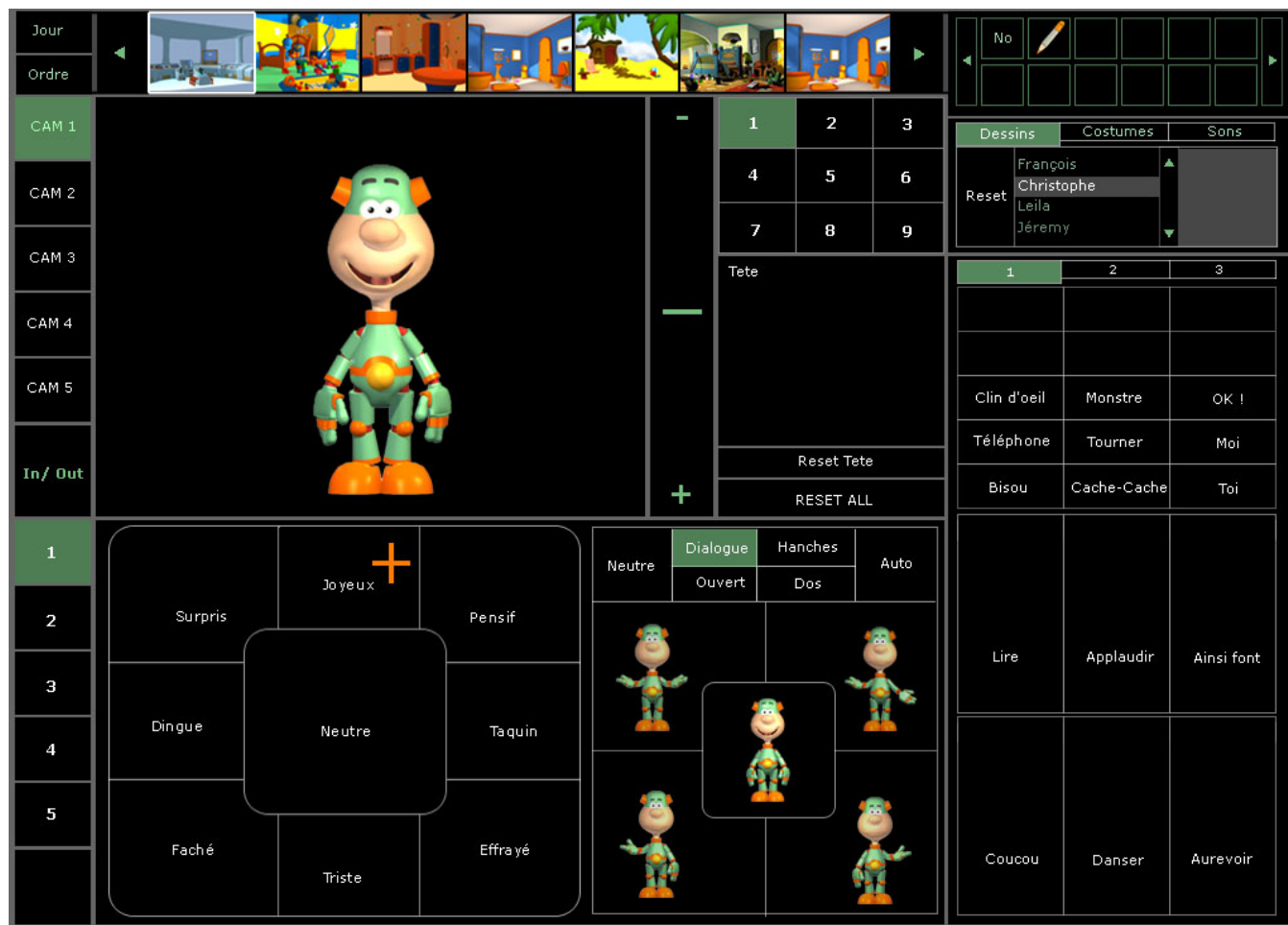


Figure 3: The animation control panel showing various emotional states that can be displayed by the virtual character.

4 A PEN-GESTURE USER INTERFACE FOR VIRTUAL PUPPET CONTROL

Fig. 3 shows a screenshot of the animation panel for controlling the virtual puppet (also shown in Fig. 1 from the first page). The user interface is composed of several interaction units: at the top, the background selection panel that sets the context for the virtual character (e.g., in the bedroom, on a planet, inside a spatial vessel); the video camera selection

panel (left) selects the hospital room and a video camera from that room; the emotion palette to convey the emotions of the virtual characters via pen gestures; and a motion panel to specify limb movements (e.g., the virtual puppet raises its hand), and predefined animations (e.g., applaud, laugh, run). A rendering of the virtual character is shown in the top-left panel for the animator. To control the virtual character, the animator can employ the following palette of gestures.

Zero-D gestures are represented by single taps by which the animator selects an emotion from the emotion palette (Fig. 3). The emotions were inspired from Fig. 2 by arranging them in a circular pattern, as follows: North (N) is happy, North-East (NE) is thinking, East (E) teasing, South-East (SE) scared, South (S) sad, South-West (SW) angry, West (W) crazy, and North-West (NW) is surprised. A tap performed in the central area of the palette returns the virtual character to its emotion-neutral state. While the animator is speaking, the virtual character shows lip movements that correspond to vowels. When an emotion is selected, a small pause between utterances inserts the specified emotion. Repeating a tap stresses the emotion displayed between consecutive utterances. A tap performed with more pressure maintains the emotional state for a prolonged time duration and with more emphasis. Taps are also used to specify the background of the virtual scene (Fig. 1, top left). Predefined animations of the virtual character, *e.g.*, applaud, walk, jump, open arms, close hands, are also executed by taps performed in distinct regions of the user interface.

1-D gestures are represented by simple flicks with various orientations that enable the animator to convey a particular emotion from the palette by ensuring smooth transition from the neutral state to that emotion. Since the palette has zones that can be crossed, the principles of crossing-based user interfaces [1] are used: as soon as the flick enters one zone, the corresponding emotion is produced. Emotions are dependent on the gesture speed and pressure controlled by the psychologist-animator: transitioning from the neutral state to an emotion zone determines the speed of the animated transition between the states of the virtual character, while the exerted pressure (with three distinct states) controls the intensity of the emotion displayed. The faster the flick gesture, the shorter the animated transition between states. The more pressure exerted, the more emphasized the emotion is. Combining speed and pressure results in a large number of options for the psychologist-animator. For example, the animator can depict a short “angry” state followed by a long “happy” one before returning to neutral.

2-D gestures are geometrical shapes exploiting the full surface of the emotion palette (Fig. 4). Gestures that render emotions (see gesture category no. 1 in Fig. 4) start from the neutral zone in the direction of any of the emotional states from the circular palette. The virtual character stays in the specified emotional state until further commands are performed. To change the emotion, the animator either selects another emotion directly (by tapping on it) or performs a gesture toward the new emotion (by means of a unistroke: see gesture examples no. 3 and 4 from Fig. 4). Gestures that start from the neutral zone, enter into an emotion zone, and then return to neutral (category no. 2 in Fig. 4) convey a

particular emotion whose duration and intensity are regulated by gesture speed and pressure. These gestures are intended to convey a transient emotion, such as being astonished, angry, or sad. When the virtual character remains in a particular emotion zone, *e.g.*, “happy,” an emotion-transition gesture (category no. 3 in Fig. 4) can be performed that starts from the current emotional zone (“happy”), traverses the central zone (“neutral”), and then enters another emotion (*e.g.*, “sad”). The speed of the gesture determines the speed of the animated transition. Unlike the gestures that specify emotional states directly, a smooth transition is ensured in this case. An emotion-transition gesture can be vertical (see gesture category no. 3 in Fig. 4) or horizontal (no. 6 in Fig. 4). Adjacent gestures represent a particular type of emotion-transition: instead of considering a target emotion, the adjacent emotion, either in the clockwise or counterclockwise direction (categories no. 5 and 6 in Fig. 4), is selected without passing through the neutral zone. Since there is no return to neutral, the transition between the two emotions is applied directly. Repeated adjacent gestures (categories no. 7 and 8 in Fig. 4) transition more than one adjacent emotional zone. For instance, a gesture can start from “happy,” go through “surprised,” and finish with “happy” again, enabling the psychologist-animator to create complex emotions.

Gesture speed was computed as the magnitude of the instant velocity between uniformly sampled points (isochronicity [19]). Linear gestures were recognized by computing the orientation of the flicks from one of the eight possible directions [7]. The gestures illustrated in Fig. 4 were implemented as unistrokes. Multistroke gestures were also recognized by turning them into unistrokes when the duration between their component strokes was under a threshold.

5 TOWARD AN EVALUATION OF THE APPLICATION FOR VIRTUAL PUPPET CONTROL AND HOSPITALIZED CHILDREN

It was not our goal in this work to perform a thorough evaluation of the user interface, but we rather focused on the technical engineering aspects of our application. Prior work has showed encouraging preliminary results for conveying emotions via cartoon-like characters [18] for the specific context of pediatric departments [8, 11], which is also our context of use. In this section, we briefly present preliminary evaluation aspects for our application:

- The cartoon-like character elicited curiosity, interest, and enthusiasm. Children became progressively attached to the virtual character to the point that some actually stole the TV remote control when they left the hospital, believing that the virtual character would be present on their TV set as well.

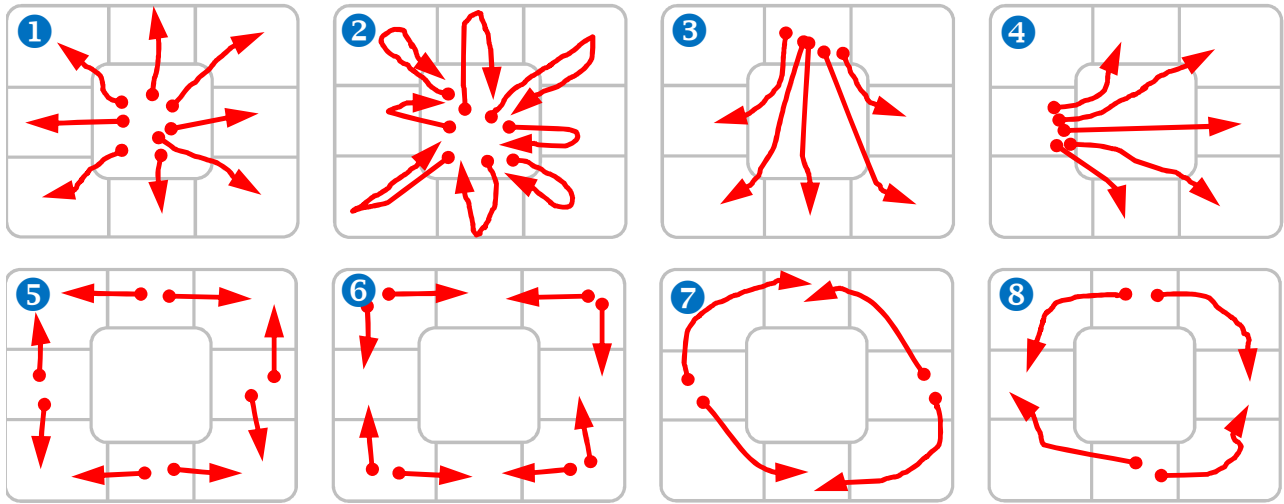


Figure 4: The gesture set used in this work to specify and control the emotions displayed by the virtual character.



Figure 5: A dragon virtual puppet may exhibit different behavior and emotions compared to a human-like character.

- The virtual character played the role of a social mediator between children and adults, e.g., their parents, the psychologist, medical personnel, etc.
- The relationship established between children with the virtual character reduced the children's stress, especially in critical conditions after surgical operations, and improved the children's perceptions of pain.
- By distracting children from the context of a hospital environment and from their own illness and by changing their moods, the virtual character made possible and facilitated medical acts which were regularly perceived as painful by children or difficult to perform on children by the medical personnel.

6 CONCLUSION AND FUTURE WORK

We presented an application and gesture-based user interface for expressing human-like emotions conveyed by a virtual character. We designed three classes of gestures to depict

individual emotions and transitions between multiple emotional states. Future work will address evaluations of our gesture set as well as of the user experience of our application. The set of gestures designed for this work is based on a simplified version of the Arousal-Valence model. We did not apply standard methodology for user-defined gestures [22, 23] to inform our set of gesture commands, but rather designed the gestures to enable fluid transitions between various emotional states arranged in a circular pattern.

Future work will also look at incorporating a larger set of emotions, but without increasing considerably the number of gesture commands to display them. While some of the gestures that could be designed for this purpose might be transferred from other contexts of use [20] (e.g., gestures for smooth transitions between emotional states and gestures showing variations in speed and pressure), user studies and experiments to evaluate the opportunity of this design option are needed.

Finally, only individual moods are conveyed through our user interface, but the ways in which they could be combined to render more complex aspects of the personality of the virtual character, according to the Hierarchical Model of Affect (Fig. 2), represent interesting directions for future work as well. For example, different moods and personalities could be attributed to different virtual characters, e.g., the dragon shown in Fig. 5 may exhibit behavior and emotions that are different from those of a human-like character.

ACKNOWLEDGMENTS

The virtual puppet depicted in this paper is Doctoon, a virtual character designed by NeuroPlanet (<http://www.doctoon.com>). Demo videos of this application are available at <https://www.youtube.com/user/Parnamont/featured>, <https://youtu.be/DGTH-XO9Bkk>, and <https://youtu.be/6FhPrd8EgBk>.

REFERENCES

- [1] Johnny Accot and Shumin Zhai. 2002. More than Dotting the i's — Foundations for Crossing-Based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. Association for Computing Machinery, New York, NY, USA, 73–80. <https://doi.org/10.1145/503376.503390>
- [2] Lisa Anthony, Radu-Daniel Vatavu, and Jacob O. Wobbrock. 2013. Understanding the Consistency of Users' Pen and Finger Stroke Gesture Articulation. In *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, CAN, 87–94.
- [3] Francesca Barrientos. 2000. Continuous Control of Avatar Gesture. In *Proceedings of the 2000 ACM Workshops on Multimedia (MULTIMEDIA '00)*. Association for Computing Machinery, New York, NY, USA, 5–8. <https://doi.org/10.1145/357744.357747>
- [4] François Beuvsens and Jean Vanderdonckt. 2012. Designing Graphical User Interfaces Integrating Gestures. In *Proceedings of the 30th ACM International Conference on Design of Communication (SIGDOC '12)*. Association for Computing Machinery, New York, NY, USA, 313–322. <https://doi.org/10.1145/2379057.2379116>
- [5] Alessandro Carcangiu and Lucio Davide Spano. 2018. G-Genie: A Gene Alignment Method for Online Partial Stroke Gestures Recognition. *Proc. ACM Hum.-Comput. Interact.* 2, EICS, Article 13 (June 2018), 17 pages. <https://doi.org/10.1145/3229095>
- [6] Céline Coutrix, Giulio Jacucci, Anna Spagnolli, Lingyi Ma, Matti Helin, Gabriela Richard, Lorenza Parisi, Stefano Roveda, and Prayag Narula. 2010. Engaging Spect-Actors with Multimodal Digital Puppetry. In *Proc. of the 6th Nordic Conf. on Human-Computer Interaction (NordiCHI '10)*. Association for Computing Machinery, New York, NY, USA, 138–147. <https://doi.org/10.1145/1868914.1868934>
- [7] Adrien Coyette, Sascha Schimke, Jean Vanderdonckt, and Claus Viehauer. 2007. Trainable Sketch Recognizer for Graphical User Interface Design. In *Proceedings of IFIP TC 13 Int. Conf. on Human-Computer Interaction (INTERACT '07)*, Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa (Eds.). Springer, Berlin, Heidelberg, 124–135. https://doi.org/10.1007/978-3-540-74796-3_14
- [8] Amélie Druez. 2005. *Approche de l'influence de DocToon sur le vécu de l'hospitalisation d'enfants de 7 à 12 ans*. Master Psychology thesis. Free University of Brussels.
- [9] Bogdan-Florin Gheran, Jean Vanderdonckt, and Radu-Daniel Vatavu. 2018. Gestures for Smart Rings: Empirical Results, Insights, and Design Implications. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18)*. Association for Computing Machinery, New York, NY, USA, 623–635. <https://doi.org/10.1145/3196709.3196741>
- [10] Sumedha Kshirsagar. 2002. A Multilayer Personality Model. In *Proceedings of the 2nd International Symposium on Smart Graphics (SMART-GRAPH '02)*. Association for Computing Machinery, New York, NY, USA, 107–115. <https://doi.org/10.1145/569005.569021>
- [11] Benoît Labaye, Nadia Guérin, and Sébastien Dohogne. 2001. DocToon® — A Mediator in the Hospital of the XXIst Century. In *Virtual Storytelling Using Virtual Reality Technologies for Storytelling*, Olivier Balet, Gérard Subsol, and Patrice Torguet (Eds.). Springer, Berlin, Heidelberg, 171–180. https://doi.org/10.1007/3-540-45420-9_19
- [12] Luís Leite. 2012. Virtual Marionette. In *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI '12)*. Association for Computing Machinery, New York, NY, USA, 363–366. <https://doi.org/10.1145/2166966.2167049>
- [13] Luis Leite and Veronica Orvalho. 2017. Mani-Pull-Action: Hand-Based Digital Puppetry. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 2 (June 2017), 16 pages. <https://doi.org/10.1145/3095804>
- [14] Hui Liang, Jian Chang, Shujie Deng, Can Chen, Ruofeng Tong, and Jianjun Zhang. 2015. Exploitation of Novel Multiplayer Gesture-Based Interaction and Virtual Puppetry for Digital Storytelling to Develop Children's Narrative Skills. In *Proc. of the 14th ACM SIGGRAPH Int. Conf. on Virtual Reality Continuum and Its Applications in Industry (VRCAI '15)*. ACM, New York, NY, USA, 63–72. <https://doi.org/10.1145/2817675.2817680>
- [15] Maya B. Mathur and David B. Reichling. 2016. Navigating a social world with robot partners: A quantitative cartography of the Uncanny Valley. *Cognition* 146 (2016), 22 – 32. <https://doi.org/10.1016/j.cognition.2015.09.008>
- [16] D. Monteiro, H. Liang, J. Wang, L. Wang, X. Wang, and Y. Yue. 2018. Evaluating the Effects of a Cartoon-Like Character with Emotions on Users' Behaviour within Virtual Reality Environments. In *Proc. of the IEEE Int. Conf. on Artificial Intelligence and Virtual Reality (AIVR '18)*. IEEE, 229–236. <https://doi.org/10.1109/AIVR.2018.00053>
- [17] Masaki Oshita. 2005. Pen-to-mime: Pen-based interactive control of a human figure. *Computers Graphics* 29, 6 (2005), 931 – 945. <https://doi.org/10.1016/j.cag.2005.09.010>
- [18] G. S. Thirunavukkarasu, H. Abdi, and N. Mohajer. 2016. A smart HMI for driving safety using emotion prediction of EEG signals. In *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC '16)*. IEEE, 004148–004153. <https://doi.org/10.1109/SMC.2016.7844882>
- [19] Jean Vanderdonckt, Paolo Roselli, and Jorge Luis Pérez-Medina. 2018. !FTL, an Articulation-Invariant Stroke Gesture Recognizer with Controllable Position, Scale, and Rotation Invariances. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction (ICMI '18)*. Association for Computing Machinery, New York, NY, USA, 125–134. <https://doi.org/10.1145/3242969.3243032>
- [20] Radu-Daniel Vatavu. 2017. Characterizing gesture knowledge transfer across multiple contexts of use. *Journal of Multimodal User Interfaces* 11, 4 (2017), 301–314. <https://doi.org/10.1007/s12193-017-0247-x>
- [21] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures as Point Clouds: A SP Recognizer for User Interface Prototypes. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI '12)*. Association for Computing Machinery, New York, NY, USA, 273–280. <https://doi.org/10.1145/2388676.2388732>
- [22] Radu-Daniel Vatavu and Jacob O. Wobbrock. 2016. Between-Subjects Elicitation Studies: Formalization and Tool Support. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3390–3402. <https://doi.org/10.1145/2858036.2858228>
- [23] Santiago Villarreal-Narvaez, Jean Vanderdonckt, Radu-Daniel Vatavu, and Jacob A. Wobbrock. 2020. A Systematic Review of Gesture Elicitation Studies: What Can We Learn from 216 Studies?. In *Proceedings of ACM Int. Conf. on Designing Interactive Systems (DIS '20)*. Association for Computing Machinery, New York, NY, USA, 25. <https://doi.org/10.1145/3357236.3395511>
- [24] Benjamin Walther-Franks, Tanja Döring, Meltem Yilmaz, and Rainer Malaka. 2019. Embodiment or Manipulation? Understanding Users' Strategies for Free-Hand Character Control. In *Proceedings of Mensch Und Computer 2019 (MuC '19)*. Association for Computing Machinery, New York, NY, USA, 661–665. <https://doi.org/10.1145/3340764.3344887>
- [25] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. Association for Computing Machinery, New York, NY, USA, 61–70. <https://doi.org/10.1145/964696.964703>
- [26] Shengdong Zhao and Mark Chignell. 2007. Using Glyphs to Facilitate Transition from Hierarchical Selection to Gesturing. *IFAC Proceedings Volumes* 40, 16 (2007), 206–211. <https://doi.org/10.3182/20070904-3-KR-2922.00036>

How Many Steps to Represent Individual Gait?

Lilian G. Motti Ader
SPHPSS, CeADAR
University College
Dublin
Dublin, Ireland
lilian.mottiader@ucd.ie

Killian McManus
SPHPSS, Insight Centre
University College
Dublin
Dublin, Ireland
killian.mcmanus@insight-centre.org

Barry R. Greene
Kinesis Health
Technologies Ltd.
Dublin, Ireland
barry.greene@kinesis.ie

Brian Caulfield
SPHPSS, Insight Centre
University College
Dublin
Dublin, Ireland
b.caulfield@ucd.ie

ABSTRACT

Assessing and reproducing user's mobility has multiple purposes for interactive systems. In particular, the quantification of gait parameters has been used for user modelling, virtual environments, and augmented reality. While many technologies can be used to assess gait, measuring spatio-temporal parameters and their fluctuations, it is important to evaluate how many steps are necessary to represent the gait pattern of an individual, in order to provide better feedback to the user and improve user experience. In this preliminary study, we evaluate the intra-session reliability of spatio-temporal gait parameters for 24 healthy adults walking two trials of 15m in a corridor. Angular velocity data were acquired from body-worn inertial measurement units attached to participants' right and left shanks. An adaptive algorithm was applied for gait event detection, and gait parameters were analyzed according to pre-defined numbers of steps extracted from the full length of the trial. The main contribution of the present analysis is to present a method of gait event detection, segmentation and analysis that can be used for adjusting interactive systems to individual users.

CCS CONCEPTS

• Human-centered computing • Human computer interaction (HCI) • HCI design and evaluation methods • User models

KEYWORDS

Gait analysis; reliability; motion-based interaction; walking

ACM Reference format:

Lilian G. Motti Ader, Killian McManus, Barry R. Greene and Brian Caulfield. 2020. How many steps to represent individual gait? In *ACM SIGCHI Symposium on Engineering Interactive*

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398638>

Computing Systems (EICS'20 Companion). June 23-26, 2020,

Sophia-Antipolis, France. ACM, New York, NY, USA, 6 pages.
<https://doi.org/10.1145/3393672.3398638>

1 Introduction

Motion-based interaction is a very exciting field of design because of advances in technologies to detect and record data from human movements. For example, mobile devices (e.g. smartphones) and wearables (e.g. smartwatches) have embedded accelerometers and gyroscopes to track device's orientation; headsets (e.g. headphones, virtual reality headsets) can adapt audio and feedback according to movements of the users' head; and sensors in the environment are used to track mobility and recognize patterns of activity.

Gait assessments have many applications for designing interactive systems. Technologies enabling gait analysis are used in areas as Health and Sports Sciences, but also in Human-Computer Interaction (HCI) in applications such as enabling for example digital human modelling for simulation [1], redirected walking for locomotion in virtual environments [2], assisted navigation and augmented reality systems [3].

Walking is the result of a fine coordination of neurological, motor and sensorial systems [4]. The use of technologies enables objective measurement of gait spatio-temporal parameters, from simple numerical values such as step counting, to more complex analysis as stride length variability. However, assessing and representing data from human movements is very challenging in terms of machine processing (e.g. data capture, treatment) and human factors (e.g. participant's abilities, morphologies, fatigue).

In addition, technologies and algorithms for gait assessment need to consider individual differences affecting pace and mobility. For example, average stride length usually depends on user's height. Users who are tall usually walk with larger stride length, travel a pre-defined distance with less steps, which can generate interaction problems (e.g. getting out of the interaction zone) as well as safety issues (e.g. hitting a wall).

Instead of relying on average values and fixed thresholds for gait event detection, algorithms can be designed to adjust event detection and provide adequate feedback to each user. For motion-based interaction, adaptive algorithms can support more realistic representation of walking and locomotion, with possible effects on segmentation of movements (e.g. bilateral gait) and comfort of use (e.g. reduction of motion sickness).

In order to improve gait assessment and representation for interactive systems, it is important to study the reliability of gait parameters, as different methods can be applied for data capture and treatment. One challenge to address is to define how much data, or how many steps, are necessary for reliable assessment of an individual's gait.

This paper presents a preliminary study on the effects of extracting pre-defined numbers of steps from the full length of a walking trial and the reliability of spatio-temporal gait parameters, calculated using a previously published adaptive algorithm [5]. Twenty-four healthy adults walked two trials of 15m in a corridor, while acceleration data was acquired from body-worn inertial measurement units attached to participants right and left shanks. The results of the intra-session reliability analysis show how the number of gait cycles included in the analysis affect the calculated measures to represent gait patterns for an individual.

2 Methods

2.1 Procedures

Participants were a sample of convenience recruited from the community and local area. A total of 24 healthy adults, mean age 30.8 ± 8 years, height 172.8 ± 8.7 cm, weight 72.2 ± 11.5 kg, 7 females (29%). Participants did not suffer from any neurological or musculoskeletal disorders. All participants provided informed consent.

Two inertial sensors were attached to each participant's shanks (about 10 cm below their knees) using dedicated Velcro straps. Each sensor contained a tri-axial accelerometer and a tri-axial gyroscope (sampling rate 102.4 Hz). Data were acquired in real time using dedicated software (Kinesis Gait™, Kinesis Health Technologies Ltd., Dublin, Ireland) and recorded via Bluetooth on a tablet device for offline analysis.

Participants were instructed to walk at a comfortable self-selected walking speed and start with their dominant foot (right or left). Each participant did two walking trials of 15m along a corridor.

2.2 Gait Events Detection

The medio-lateral angular velocity is recorded from right and left sensor's gyroscopes, as specified in the literature [6]. After calibration, raw gyroscope signal represents the shank angular velocity. Calibration, data treatment and artefact rejection of sensor data is reported elsewhere [5]. Detection of gait events and calculation of spatial and temporal parameters follow the procedures described in published algorithms [5,7,8].

Figure 1 represents gait event detection and segmentation on a sample of shank angular velocity.

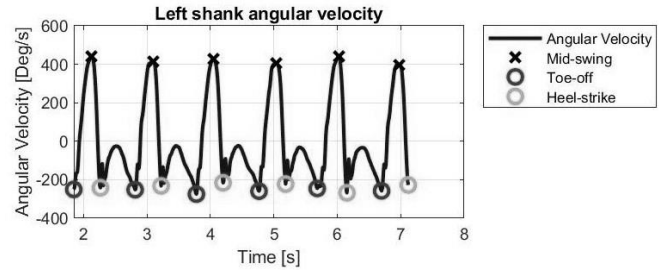


Figure 1: Plot representing gait event detection from left medio-lateral shank angular velocity and six strides of steady state gait extracted from the full length of the walking task for one participant (female, 34 years old, height 165 cm)

The analysis of the medio-lateral shank angular velocity signal allows the identification of heel strike (initial contact) and toe-off (terminal contact) times representing the mobilization of each leg. From these events, we can estimate the sequence of events for a bipedal gait cycle and calculate gait spatio-temporal parameters:

- Stance time: heel strike to toes-off of the same foot (mean times, averaged across both legs, in seconds)
- Swing time: the foot is not in contact with the ground between toes-off and heel strike (mean times, averaged across both legs, in seconds)
- Stride time: time between successive heel strikes of the same foot, corresponds to a gait cycle (mean times, averaged across both legs, in seconds)
- Step time: heel strike of one foot to heel strike of opposite foot (average of times, in seconds)
- Double support: proportion of gait cycles spent on both feet (averaged across multiple gait cycles, expressed as percentage)
- Stride length: distance travelled during swing time estimated from orientation angle of the shank relative to the vertical, considering the participant's height (average value in meters)
- Stride velocity: stride length divided by stride time (cm/s), averaged across both legs.

To estimate the variability, the Coefficient of Variation (CV) of the spatio-temporal gait parameters was used calculated as the standard deviation (SD) divided by mean values within subject across multiple gait cycles, expressed as a percentage.

Figure 2 illustrates the gait cycles phases.

How many steps to represent individual gait?

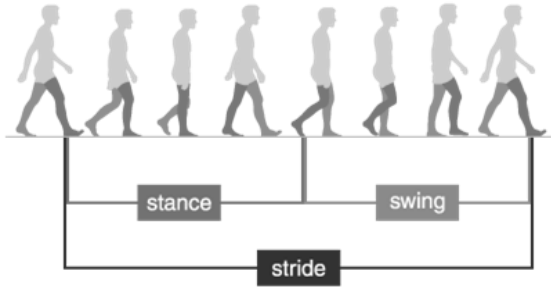


Figure 2: Chart illustrating gait cycles phases (Reproduced with permission from Kinesis Health Technologies © Source: <https://www.kinesis.ie/gait/>)

2.3 Data Segmentation

Gait assessments should consider the different phases of a walking trial: gait acceleration or initiation phase, steady state gait and deceleration or termination phase. For the purposes of the present study, the analysis focuses on steady state gait data. To remove gait initiation phase, the segmentation of gait data ignored the first gait cycle (i.e. first step each leg) and included data from the 3rd step, as commonly reported in the literature [9,10].

Parameters relying on data from both legs can only be estimated from a minimum of two gait cycles. Variability of temporal parameters and spatial parameters requires at least three gait cycles. Participants executed different number of steps to complete the trials. In order to include all the participants, a maximum of seven gait cycles was extracted from the total length of the trial (14 strides, or seven strides per leg). Seven gait cycles represent the total steady state gait for the walking trial in the present study.

2.4 Data Analysis

The intra-session reliability was calculated using intraclass correlation coefficients (ICC(2,k)) [11,12] for each one of the gait parameters, for a pre-defined number of gait cycles, ranging from 2 to 6. The results were classified according to the ICC scores indicating reliability is excellent ($ICC \geq 0.90$), high ($0.75 \leq ICC < 0.90$), moderate ($0.50 \leq ICC < 0.75$) or poor ($ICC < 0.50$) [11]. Data analysis was conducted offline using MATLAB (version R2019a, MathWorks, VA, USA).

3 Results

3.1 Intra-session reliability

Spatio-temporal gait parameters calculated from mean values present excellent reliability from two gait cycles (four strides, two of each leg). Figure 3 presents reliability of spatio-temporal gait parameters according to the number of gait cycles.

EICS'20 Companion, June 23-26,2020, Sophia Antipolis, France

Reliability of parameters related to the variability tend to increase with the number of gait cycles included in the analysis. Most parameters describing gait variability reach moderate reliability from five gait cycles and good reliability when six gait cycles or more are included in the analysis. Figure 4 presents reliability of variability of temporal gait parameters according to the number of gait cycles.

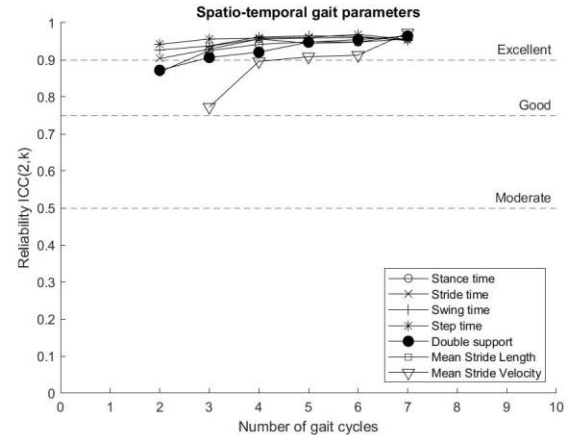


Figure 3: Reliability of spatio-temporal gait parameters according to the number of gait cycles included in the analysis

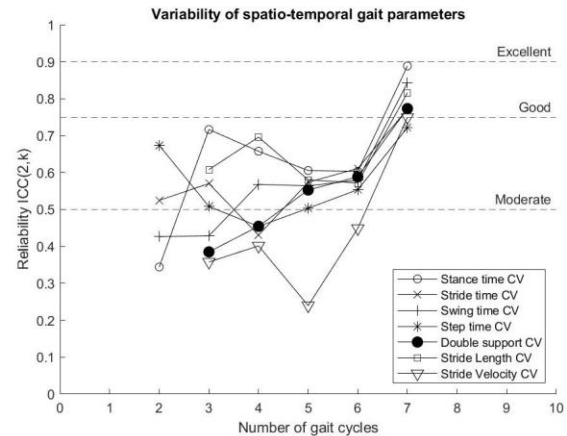


Figure 4: Reliability of the variability of spatio-temporal gait parameters according to the number of gait cycles included in the analysis

4 Discussion and Conclusion

The present study analyzed the intra-session reliability of spatio-temporal gait parameters and their variability, according to a pre-defined number of gait cycles representing steady-state gait and extracted from the full walking trial. In the present study, data collected with body-worn inertial sensors from participants' right and left shanks were calculated after an adaptive algorithm for gait event detection.

Results show that the selected algorithm for adjusting gait event detection to each individual can obtain reliable measures for parameters based on averaged values from 3 gait cycles (6 steps, 3 from each leg). The reliability of parameters representing gait variability increases with the number of steps included in the analysis and values might need to be readjusted for a better representation of gait in long walking tasks.

This preliminary study showed that the combination of the selected adaptive algorithm, extracting gait events from data collected from the right and left shanks, and a windowed approach, extracting a pre-defined number of gait cycles from the full length of the walking task, should be explored with a larger number of participants to further evaluation of the reliability of measures of individual gait assessments.

The gait signature, or the individual characteristics of walking, is the result of the fine tuning of neuro, motor and sensorial systems, ensuring postural stability and balance control. We expect that adaptive algorithms can enable more realistic representation of walking and locomotion, improving the correspondence of real and virtual displacements.

ACKNOWLEDGMENTS

L.G.M.A. receive funding from the EU H2020 under the Marie Skłodowska-Curie Career-FIT fellowship (Co-fund grant No. 713654). K.McM. is an employee at Kinesis Health Technologies Ltd, as part of the Irish Research Council Employment-Based Post-Graduate Programme. B.R.G. is a director of Kinesis Health Technologies Ltd, a company with a license to commercialize Kinesis Gait™.

REFERENCES

- [1] N. Kaklanis, P. Biswas, Y. Mohamad, M.F. Gonzalez, M. Peissner, P. Langdon, D. Tzovaras, C. Jung, Towards standardisation of user models for simulation and adaptation purposes, *Univers. Access Inf. Soc.* 15 (2016) 21–48.
<https://doi.org/10.1007/s10209-014-0371-2>.
- [2] J. Walker, Redirected Walking in Virtual Environments, *Cs.Mtu.Edu.* (2013) 1–14.
<http://cs.mtu.edu/~jwwalker/files/cs5760-jwwalker-topicpaper.pdf>.
- [3] M. Sra, A. Mottelson, P. Maes, Your place and mine: Designing a shared VR experience for remotely located users, *DIS 2018 - Proc. 2018 Des. Interact. Syst. Conf.* (2018) 85–98.
<https://doi.org/10.1145/3196709.3196788>.
- [4] J.M. Hausdorff, Gait variability: methods, modeling and meaning Example of Increased Stride Time Variability in Elderly Fallers Quantification of Stride-to-Stride Fluctuations, 9 (2005) 1–9.
<https://doi.org/10.1186/1743-Received>.
- [5] B.R. Greene, D. McGrath, R. O'Neill, K.J. O'Donovan, A. Burns, B. Caulfield, An adaptive gyroscope-based algorithm for temporal gait analysis, *Med. Biol. Eng. Comput.* 48 (2010) 1251–1260.
<https://doi.org/10.1007/s11517-010-0692-0>.
- [6] E.P. Doheny, T.G. Foran, B.R. Greene, A single gyroscope method for spatial gait analysis, 2010 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC'10. (2010) 1300–1303.
<https://doi.org/10.1109/IEMBS.2010.5626397>.
- [7] B.R. Greene, T.G. Foran, D. McGrath, E.P. Doheny, A. Burns, B. Caulfield, A comparison of algorithms for body-worn sensor-based spatio-temporal gait parameters to the gaitrite electronic walkway, *J. Appl. Biomech.* 28 (2012) 349–355.
<https://doi.org/10.1123/jab.28.3.349>.
- [8] U. Lindemann, B. Najafi, W. Zijlstra, K. Hauer, R. Muche, C. Becker, K. Aminian, Distance to achieve steady state walking speed in frail elderly persons, 27 (2008) 91–96.
<https://doi.org/10.1016/j.gaitpost.2007.02.005>.
- [9] K.S. Van Schooten, S.M. Rispens, P.J.M. Elders, J.H. Van Diee, Toward ambulatory balance assessment: Estimating variability and stability from short bouts of gait, 39 (2014) 695–699.
<https://doi.org/10.1016/j.gaitpost.2013.09.020>.
- [10] T.K. Koo, M.Y. Li, A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research, *J. Chiropr. Med.* 15 (2016) 155–163. <https://doi.org/10.1016/j.jcm.2016.02.012>.
- [11] K.O. McGraw, S.P. Wong, “Forming inferences about some intraclass correlations coefficients”: Correction., *Psychol. Methods.* 1 (1996) 390–390.
<https://doi.org/10.1037//1082-989x.1.4.390>.

SeqCheck: A Model Checking Tool for Interactive Systems

Jessica Turner

jessica.turner@waikato.ac.nz
University of Waikato
Tauranga, New Zealand

Judy Bowen

judy.bowen@waikato.ac.nz
University of Waikato
Hamilton, New Zealand

Steve Reeves

steve.reeves@waikato.ac.nz
University of Waikato
Hamilton, New Zealand

ABSTRACT

In this paper we describe SeqCheck, a model checking tool which allows us to investigate if certain properties hold for an interactive system. These properties allow us to determine if the interaction sequence model of an interactive system's overlap component behaves as expected. We describe the properties we have defined for this overlap component and then demonstrate the use of SeqCheck to identify when these do not hold.

CCS CONCEPTS

• **Software and its engineering** → **Formal methods**; *Model-driven software engineering*.

KEYWORDS

Interactive Systems; Interaction Sequences; Model Checking; Finite State Automata.

ACM Reference Format:

Jessica Turner, Judy Bowen, and Steve Reeves. 2020. SeqCheck: A Model Checking Tool for Interactive Systems. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398639>

1 INTRODUCTION

Interactive systems are software systems or hardware devices with a user interface that requires human input. In this research interactive systems are defined conceptually as having three components: the interactive, functional and overlap. The interactive component is the user interface of the system and is made up of interactive elements (widgets).

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398639>

The functional component performs the necessary instructions and calculations triggered by the end-user interacting with the user interface. The overlap component is the point at which the functional and interactive components intersect. Problems can arise in this intersection, and it is therefore necessary to ensure it behaves as expected. This conceptual separation is commonly seen in the Model View Controller (MVC) pattern (see [9, 14]). For example, we can consider the model as the functional component, the view as the interactive component, and the controller as the overlap component.

In an MVC pattern it is obvious that if the controller fails to behave as expected, a direct consequence is that the model and view will also fail to behave as expected. Similarly, if the overlap component does not behave as expected, the interactive and functional components cannot communicate and this results in unexpected behaviours which could lead to error. In safety-critical situations this unexpected behaviour could lead to harm to the end-user or even fatalities. This highlights the importance of ensuring the overlap component works as expected and the motivation for this work.

In previous research the self-containment property was introduced by Turner *et al.* [15, 17] which supports the modelling of interactive systems using interaction sequences. Turner *et al.* adapt these self-contained interaction sequences to inform a model-based testing approach [15]. In this work first we define a set of properties (the overlap properties) which must be satisfied to ensure the overlap is defined correctly. In [16] Turner *et al.* describe the Sequence Simulator, a tool which allows for the simulation of interaction sequences, we adapt this tool as the basis for the SeqCheck model checker¹. The main contribution of this work is that we present both the overlap properties and the SeqCheck tool which allows us to automatically determine if the properties hold for the specified models.

2 THE OVERLAP PROPERTIES

In this section we describe the properties required to ensure the overlap component works as expected. In general, the overlap properties hold if the interactive and functional

¹Available from <https://github.com/jdt11/SeqCheck>

components produce the required result based on a set of actions.

We can think of the interactive component as having N widgets (where N is the total number of widgets). An interaction (action) with each of these widgets “triggers” an instruction which is conveyed to the functional component via the overlap component. A trigger occurs when the end-user interacts with a widget in order to perform some action. Similarly, the functional component has N functions (where N is the total number of functions). Which set of functions to execute is communicated to the functional component by the overlap component via the triggered instruction.

For example, for a button widget called “HelloWorld!” an action a user may wish to perform is “click”. This action would trigger an instruction which tells the functional component to trigger the function “Display Hello”. This requires a modification of a display widget in the interactive component, this widget must display the message generated by the “Display Hello” function. Once the functional component has finished executing the function, it sends a message to the display. This type of relationship is captured by the first overlap property and is defined generally as follows:

Definition 2.1 (General Overlap Property 1, GOP1). An interactive system has GOP1 iff an action triggered in that system triggers all corresponding behaviours (and no others), in the interactive or functional component.

Definition 2.1 states that in the model of the system each action triggers only the expected behaviour. This is important in terms of interaction sequences because if a single widget does not trigger the expected behaviour a sequence which includes that widget will have unexpected results i.e. it will not behave as expected.

After a function has been triggered it may trigger several other functions which finally returns a result. This result usually needs to be communicated to the end-user, and is therefore returned to the interactive component via the overlap component, again in the form of an instruction. However, these instructions are different from the previously described triggers, in that they may contain information which allows the interactive component to be modified. It is important that the correct information is returned to the appropriate widget so that this information can be conveyed to the end-user correctly. More importantly, end-users often send a set of triggers in the form of an interaction sequence in order to achieve some goal or complete some task, therefore we must ensure not only that a single widget trigger returns the correct information but that combinations of triggers return the correct result to the interactive component.

In previous work [15–17] Turner *et al.* explored ways in which to constrain interaction sequences so that they would remain tractable for testing purposes. This results in

grouping the sequences via tasks using the self-containment property. Each interactive system may have a set of tasks $T = \{T_1, T_2, \dots, T_n\}$, and for each task there are associated interaction sequences. Interaction sequences are defined as having a start state, followed by a set of actions, which result in an expected end state. The start and end states are described as a set of observations from the functional component.

For example, a task T_i has a set of triples $(s_i, a_i, e_i) \in T_i$ where $1 \leq i \leq n$ such that n is the number of triples, s_i is a set of starting observations, a_i is a sequence of actions (or interaction sequence) which results in the expected end state e_i . Therefore, for an interactive system which has property GOP1 it must also have the following in order for the overlap component to behave as expected for each task.

Definition 2.2 (General Overlap Property 2, GOP2). An interactive system (that meets GOP1) has GOP2 iff the end state that follows from the start state of the system and a sequence of actions matches the required end state.

For the overlap properties to be satisfied definition 2.2 must be satisfied for each task in T . However, we may wish to investigate only a subset of the tasks in T , therefore we may restrict T to the tasks which are of interest. For example, in interactive systems there may be tasks which are implemented using pre-existing packages and/or libraries. As these cannot be modified or improved there is no benefit in exploring these tasks, as a result they can be removed from T . Furthermore, there may be a subset of tasks which are safety-critical. We may wish to only explore those safety-critical tasks, as errors in these tasks can have more significant consequences.

3 MODELS

In this research Finite State Automata (FSA) [6] are used to model interaction sequences. An automaton represents an interaction sequence in terms of its widgets and interactions. We can create an automaton for each task of the system and use the FSA to further explore sequences of varying lengths for that specific task. This is useful as errors typically occur in longer sequences.

The automaton of the sequence is modelled in combination with a set of observations which represent the start state before the sequence is executed and the end state after the sequence has taken place. The observations come from the observations of the functional component and the current state of the interactive component. These observations allow us to clearly specify what changes should take place after the sequence has triggered a specific set of actions.

In order to explore the overlap component we also require models of the interactive and functional components. To model the interactive component we use Presentation

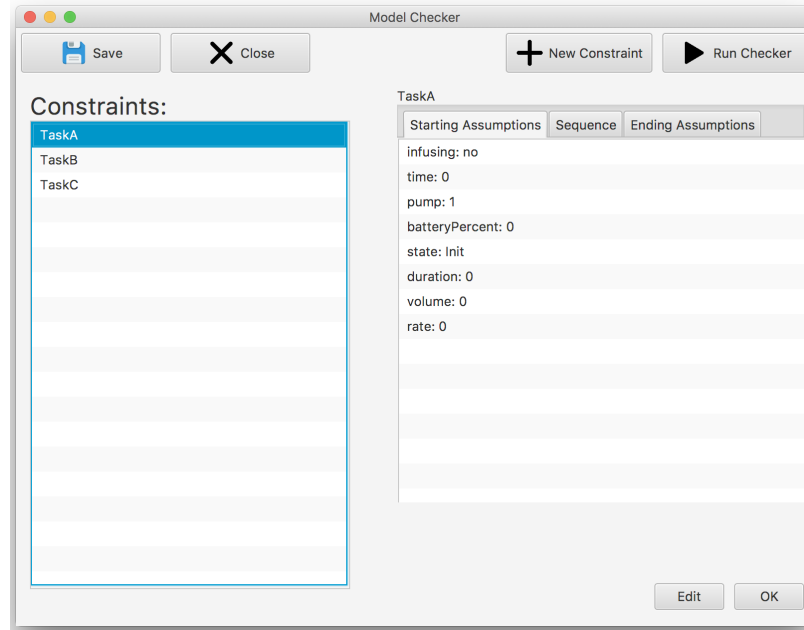


Figure 1: The SeqCheck Model Checker

Models (PModels), Presentation Interaction Models (PIMs), and Presentation Model Relations (PMRs) [1]. PModels describe the behaviour of an interactive system in terms of its interactive components. PIMs define the behaviours which allow us to navigate between different modes or states of the interactive system. Lastly, A PMR is a relation which defines behaviours which relate to a formal specification of the system.

The functional component of the interactive system is defined as a formal specification, in this case we have chosen the Z language [7, 18]. A Z specification describes the changes to the observations of a system with regards to the state space. The Z language was chosen as it is easy to integrate with PModels and PIMs through the use of the PMR. Furthermore, we were able to use the ProB Model Checker Tool [8, 10] to simulate the behaviour of the Z specification in order to implement our model checker for the overlap component.

In [16] Turner *et al.* describe a proof-of-concept tool, the Sequence Simulator, which allows for the simulation of interaction sequences using the models that we describe here. The SeqCheck tool uses some of the concepts explored in the Sequence Simulator to allow for the input and exploration of sequences. However, the focus here is to use our defined overlap properties to show that the overlap component behaves as expected, as opposed to the proof-of-concept simulation in [16].

In order to use the overlap properties with the models we have described we specialised each property formally for those models. These properties are then used by SeqCheck to ensure that the overlap properties hold for specific tasks. In the next section we describe the SeqCheck tool and how it uses each of these models to determine if the properties hold. We also describe how the tool clearly communicates where problems occur.

4 SEQCHECK: THE MODEL CHECKER TOOL

The SeqCheck Tool is depicted in figure 1. In order to use the model checker we require models for each of the different interaction components.

The PModels, PIM and PMR are created using the PIMed Tool² which allows the user to save the file in an XML format. The Z Specification is created using the ProB Animator and Model Checker (ProB)³. The ProB tool allows us to explore the state space of the functional specification and also allows for model checking, however, this is only of the functional specification. In our tool we use this functionality of the ProB tool in order to allow us to explore the functional specification in relation to the overlap component. The FSA file, or rather the file for the interaction sequence model, may be created directly in the SeqCheck tool when starting a new project.

²See <https://sourceforge.net/projects/pims1/>

³See https://www3.hhu.de/stups/prob/index.php/Main_Page

In order to determine if the overlap properties hold with these models we required a way in which to explore each of their state spaces that were relevant to the overlap component. The functional specification was easy to explore as it could be determined using the ProB tool, which required minor expansion to connect to the SeqCheck tool. However, the PModels, PIM, PMR and FSA could not be considered using the ProB tool. The Sequence Simulator tool described in [16] already allows for the simulation of interaction sequences and in some part allows us to explore the state space of these other models. This functionality was extended in the SeqCheck tool in order to allow us to directly investigate the overlap properties.

As described in section 2 an interactive system has a set of tasks T such that $T = \{T_1, T_2, \dots, T_n\}$, and for each task in T there exists a triple $(s_i, a_i, e_i) \in T_i$ where s_i is a set of starting observations, a_i is a sequence of actions (or interaction sequence) which results in the expected end state e_i . In the SeqCheck Tool we refer to each triple (s_i, a_i, e_i) for a specific task as a constraint. Typically each constraint is given a meaningful name such as “IncreaseVolTask” and they are displayed in a list which allows the user to inspect the values of each part of the triple, the “Starting Assumptions” s_i , the “Sequence” a_i , and the “Ending Assumptions” e_i (see figure 1). Each of these can be edited or modified as required.

If the user wishes to add a new constraint $Task_A$ they are first presented with a set of labels and default inputs for each of the observations. They are also presented with a state observation which refers to states specified in the PIM. The user can specify the relations for s_{Task_A} by specifying the values for each observation.

Next they can enter a sequence by selecting from a set of “next” possible steps. This input presents the user with all of the next available triples in the finite state automaton beginning with the start state. This allows the user to build an interaction sequence (a_{Task_A}) following the specified interaction sequence model.

Lastly, similarly to the starting assumptions, we can specify e_{Task_A} by specifying values for the observations. The combination of this creates the triple for $Task_A$ such that $Task_{A_i} = \{(s_{Task_A}, a_{Task_A}, e_{Task_A})\}$. This constraint is added to the existing list of constraints. The user may repeat this process until they have added the desired number of constraints for this task.

In order to determine if the overlap properties hold for the specified tasks, the user must select the “Run Checker” button. This triggers the system to automatically check the properties on the models which are currently loaded in the system. A detailed report is provided to the user which details where discrepancies were found. An example of one of these reports is given in listing 1.

Listing 1: An example of The Model Checker Tool output

```

1  --- Starting constraints ---
2  The widgets match.
3  The S-behaviours match.
4  The Z schemas match.
5  --- 1. TaskA ---
6  Size of provided observations: 8
7  Size of observations: 8
8  Observation set size is equal.
9  Found observation infusing; values are
   equal (infusing=no).
10 ...
11 The states match.
12 Step: (Initialise, Click, YesStartButton
   )
13 Behaviours are: [I_SetVolume]
14 Next state is: SetVolume
15 Completed behaviour: I_SetVolume
16 Step: (YesStartButton, Observe, Display)
17 ...
18 Errors found in starting observations:
   0.
19 Errors found in ending observations: 0.
20 Total Errors in observations: 0.
21 ---
22
23 --- 2. TaskB ---
24 ...
25 Found observation pump; values are not
   equal.
26 pump=100 and pump=1.
27 ...
28 Errors found in starting observations:
   1.
29 Errors found in ending observations: 1.
30 Total Errors in observations: 2.
31 ...

```

In listing 1, lines 2-8 provide the output from checking if GOP1 holds. This is performed by ensuring that the widgets exist in the PModel and the Interaction Sequence Model (the automaton), that the behaviours in the PMR match those specified in the PModel, and that the specifications that they map to exist in the Z specification. Lastly, the observations are checked to match those provided in the constraint and that they ensure they are of the same number as those in the model. This is performed to ensure that before we model check GOP2 the observations are equal, otherwise there is no

point to continue with the task, as GOP1 is not satisfied. As we can see for this example all of these checks are successful.

Next, we move on to determining if GOP2 holds or not for each of the specified constraints (tasks). This involves ensuring that the start state is correct, performing the sequence specifying which behaviours are triggered, and finally ensuring that the end state is as expected. The last output given is the number of errors found in the model. Note that feedback is also given at any point in the sequence where the check fails.

On lines 25-30 of listing 1 we can see an example of an end state observation which is not as expected. This is reported to the user and added to the count of errors. This clearly shows where the overlap property is not satisfied. In this example while GOP1 holds, GOP2 does not, and therefore the overlap component is not as expected. This means that while each of the widgets trigger the correct instructions, the combination of the instructions does not produce the expected behaviour. At this point, the end-user may diagnose and implement solutions to correct this behaviour and re-run the model checker to determine if the implemented solution is correct.

5 RELATED WORK

The ProB model checker is used by Leuschel *et al.* and Körner *et al.* [8, 10] to produce counter-examples when exploring B specifications. They then extended this to include further specification languages. The counter-examples produced are in the form of paths through the specification which is represented as an automaton. While the SeqCheck tool does not specifically produce counter-examples at this stage we can identify counter-examples using the overlap properties. In future work, the SeqCheck tool could be extended to generate and find counter-examples.

In [2] Brat *et al.* describe a technique for the verification and validation of interactive systems, specifically using the Petshop and Java PathFinder tools. In this paper they describe an approach which allows them to produce properties to test on the model of the interactive system based on the system's needs and requirements. This is different to the properties described here, as the properties defined by Brat *et al.* are at a component level rather than a system level (those which hold for all interactive systems).

Loer *et al.* [11] discuss the difficulty in using model checking systems as properties are often expressed in formal logic. Furthermore, they state that the counter-examples produced may also be difficult to understand for the same reason. In the SeqCheck Tool, due to the way the overlap properties have been specified, we are able to remove this complexity from the user. This is the reasoning behind providing descriptive textual feedback to the user in order to allow them to more easily determine where properties do not hold.

PVSIO-web is a tool created to prototype and help check the safety of medical device interfaces [12, 13], by visualising the formal verification of a system in PVS. In [12, 13] Masci *et al.* have a clear focus on the interactive component to ensure that it behaves as expected. The graphical display and simulation of the underlying functional model is similar to how we explore the formal models for each of the interactive system components in the SeqCheck tool, however, at this stage we do not display this simulation visually.

In [3–5] the authors describe the use of the IVY workbench in order to perform formal analysis on user interfaces for safety-critical interactive systems. In [5] Couto *et al.* specifically discuss improvements to the tool, to allow for easier understanding of the counter-examples found. Similarly to PVSIO-web and Petshop with Java PathFinder the tool focuses on the interactive component as opposed to the overlap component.

6 FUTURE WORK AND CONCLUSIONS

In this paper we discuss the overlap properties for an interactive system and introduce the SeqCheck tool which allows us to automatically determine if these properties hold for a given interactive system. We described the general overlap properties and explained the different models required to use the tool, identifying which model represented which component of the interactive system. Next we described the SeqCheck tool and looked at some of the example output provided after model checking has occurred. Lastly, we discussed other similar tools to position our work within the larger set of model-checking tools for interactive systems.

Note that model-checking the overlap properties is not sufficient alone in order to determine the correctness of an interactive system. The reason for this being that the interactive and functional components must also be checked. However, as there are well established and existing techniques which allow for this, such as those discussed in section 5, we did not explore using the models for these purposes.

In future iterations of the SeqCheck tool it would be beneficial to include a way to explore the properties and counter-examples graphically, in order to ensure better understanding and reduce the learning curve for model-checking. This would allow for easy exploration and a more “natural” way of inputting task constraints, as a user would be able to do so visually. However, user interfaces have a tendency to change over time as the system is improved, therefore, it is essential that any technique implemented allows for these types of updates.

REFERENCES

- [1] J. Bowen and S. Reeves. 2013. UI-design Driven Model-based Testing. *Innovations in Systems and Software Engineering* 9, 3 (2013), 201–215. <https://doi.org/10.1007/s11334-013-0199-6>

- [2] G. Brat, C. Martinie, and P. Palanque. 2013. V&V of Lexical, Syntactic and Semantic Properties for Interactive Systems through Model Checking of Formal Description of Dialog. In *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments*, M. Kurosu (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 290–299.
- [3] J. C. Campos, M. Sousa, M. C. B. Alves, and M. D. Harrison. 2016. Formal Verification of a Space System’s User Interface With the IVY Workbench. *IEEE Transactions on Human-Machine Systems* 46, 2 (April 2016), 303–316. <https://doi.org/10.1109/THMS.2015.2421511>
- [4] J. C. Campos and N. Sousa. 2018. The MAL Interactors Animator: Supporting Model Validation through Animation. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '18)*. Association for Computing Machinery, New York, NY, USA, Article Article 11, 7 pages. <https://doi.org/10.1145/3220134.3220142>
- [5] R. Couto and J. C. Campos. 2019. IVY 2: A Model-Based Analysis Tool. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 5, 6 pages. <https://doi.org/10.1145/3319499.3328228>
- [6] J. E. Hopcroft. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, Mass.
- [7] J. Jacky. 1997. *The Way of Z: Practical Programming with Formal Methods*. Cambridge University Press, Cambridge ; New York, NY, USA.
- [8] P. Körner and J. Bendisposto. 2018. Distributed model checking using ProB. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10811, 244–260.
- [9] Glenn E. Krasner and Stephen T. Pope. 1988. A Cookbook for Using the Model-view Controller User Interface Paradigm in Smalltalk-80. *J. Object Oriented Program.* 1, 3 (Aug. 1988), 26–49. <http://dl.acm.org/citation.cfm?id=50757.50759>
- [10] M. Leuschel and M. Butler. 2003. ProB: A Model Checker for B. *Lecture Notes in Computer Science* 2805.
- [11] K. Loer and M. Harrison. 2002. Towards usable and relevant model checking techniques for the analysis of dependable interactive systems. In *Proceedings 17th IEEE International Conference on Automated Software Engineering.*, 223–226. <https://doi.org/10.1109/ASE.2002.1115016>
- [12] P. Masci, P. Oladimeji, P. Curzon, and H. Thimbleby. 2017. Using PVSio-web to Demonstrate Software Issues in Medical User Interfaces. In *Software Engineering in Health Care*, M. Huhn and L. Williams (Eds.). Springer International Publishing, Cham, 214–221.
- [13] P. Masci, P. Oladimeji, P. Mallozzi, P. Curzon, and H. Thimbleby. 2016. PVSio-web: mathematically based tool support for the design of interactive and interoperable medical systems. *EAI Endorsed Transactions on Collaborative Computing* 2, 7 (11 2016), 42–45. <http://ezproxy.waikato.ac.nz/login?url=https://search-proquest-com.ezproxy.waikato.ac.nz/docview/2306216856?accountid=17287>
- [14] C. Romanyk, R. McCallum, and P. Salehi. 2016. A Model Based Approach to Web Application Design for Older Adults using MVC Design Pattern. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9752. Springer Verlag, 348–357.
- [15] J. Turner. 2019. *Supporting Interactive System Testing using Interaction Sequences*. Ph.D. Dissertation. The University of Waikato, Hamilton, New Zealand.
- [16] J. Turner, J. Bowen, and S. Reeves. 2018. Simulating Interaction Sequences. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '18)*. ACM, New York, NY, USA, Article 8, 7 pages. <https://doi.org/10.1145/3220134.3220139>
- [17] J. Turner, J. Bowen, and S. Reeves. 2018. Using Abstraction with Interaction Sequences for Interactive System Modelling. In *Software Technologies: Applications and Foundations*, M. Mazzara, I. Ober, and G. Salaün (Eds.). Springer International Publishing, Cham, 257–273.
- [18] J. Woodcock and J. Davies. 1996. *Using Z : specification, refinement, and proof*. Prentice Hall, Hemel Hempstead, Herts.

Development Framework for Context-Aware Augmented Reality Applications

Sarah Krings, Enes Yigitbas, Ivan Jovanovikj, Stefan Sauer, Gregor Engels

Paderborn University, Germany, {skrings@mail., enes@mail., ivanj@mail., sauer@, engels@}upb.de

ABSTRACT

Augmented Reality (AR) is a technology on the rise. Due to its growing popularity and application in various domains, ways of increasing user friendliness and usability of AR are now becoming more important. Context-awareness is one of these ways, as it can make an AR application adjust to the user, their situation and their needs, making the application more ergonomic and easy to work with. Especially in mobile AR, context changes happen often, due to the user moving around and using their devices in different environments and situations. In order to ease the development of context-aware applications for mobile AR, the modular development framework AARCon is proposed. AARCon consists of context monitoring features to keep track of the context of use and adaptation features to react to context changes. To show the potential of AARCon, a case study is presented, where AARCon is used to create a context-aware AR printer maintenance application.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**.

KEYWORDS

context-aware augmented reality; context-aware AR development framework

ACM Reference Format:

Sarah Krings, Enes Yigitbas, Ivan Jovanovikj, Stefan Sauer, Gregor Engels. 2020. Development Framework for Context-Aware Augmented Reality Applications. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398640>

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398640>

1 INTRODUCTION

Augmented Reality (AR) has been researched for a considerable amount of time [9] [4], but has only gained increased public attention in more recent years. Especially recent approaches in mobile AR are quite popular, which is reflected in the success of smartphone applications like Niantic's Pokémon GO¹ with over 100 million downloads or Ikea's IKEA Place², which was downloaded at least 100.000 times.

But AR has more facets than just displaying virtual furniture in the real world. AR is commonly defined as a combination of real and virtual contents, which is interactive in real time and registered in 3D [2]. Due to high technical proliferation, AR can be used in many different scenarios and contexts. Mobile AR (AR on mobile devices like smartphones) also offers many new surroundings for the use of AR due to usage in varying context-of-use situations. This occurs even more in approaches like pervasive AR [5], an approach where AR is used continuously, not just in different specific contexts. In use cases like the ones mentioned here, a lot of context changes happen. These are changes in anything that might affect the application, for example changes concerning the environment, the user or the device the application is running on [3]. An application adjusts to this context by adapting itself to better suit the current context is called context-aware [1]. Context-awareness can be of advantage, for example it can increase usability by adjusting to every user or to new surroundings, taking the burden and distraction of making these adjustments away from the user.

In conclusion, the topic of context-aware AR is especially important for future pervasive technology and is an advantage for usage of AR applications. Furthermore, AR in general is a future technology on the rise which offers a lot of possibilities and has a fast-growing audience, especially in AR for handheld smartphones. However, there are only few approaches introducing context-awareness to AR yet and they are mostly limited to a specific field. There are more general approaches, but these mostly do not include AR-specific features (e.g. distance to virtual objects or environment lighting). A more broad approach is needed to support the use of context-awareness in AR in general. In this work, we introduce AARCon, short for "Android-based framework for

¹ <https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo>

² https://play.google.com/store/apps/details?id=com.inter_ikea.place

Augmented Reality with Context-Awareness”. It supports the development of context-aware AR in a modular, extensible manner to make it possible to use our approach in a broad range of scenarios. The remainder of the paper is structured as follows. We give an overview of the AARCon framework in Section 2 before we move on to the implementation details in Section 3. To show the feasibility of our approach, in Section 4, we present a case study where AARCon was used to develop a context-aware printer maintenance application. In Section 5, we present the currently available approaches and compare them to AARCon. Finally, in Section 6, we conclude our work and give a brief outlook on future work.

2 THE AARCON FRAMEWORK

In this section, we introduce the AARCon framework. As illustrated in Figure 1, it consists of three main components: *Context Monitoring*, *Adaptation*, and *Decision Making*.

The *Context Monitoring* component on the top left is responsible for constantly collecting information about different kinds of context to enable the framework to react to them appropriately. All the information on the context is measured by sensors; partially real sensors like the phone camera or the inertial sensors like gyroscopes, partially in a more figurative sense like measuring information about the app such as positioning of virtual objects or usage. The sensor data is read out through *Sensor APIs* and used by several subcomponents which are each responsible for checking on one specific context feature (e.g. *DarkCondition* and *DistanceToUserBigCondition* in Figure 1). The entirety of the *Context Monitoring Features* is used by the *Condition* and each of the context features has its own component responsible for monitoring it. The specific procedures and connections will be elaborated on in Section 3. The *Decision Making* (on the right side of Figure 1) is led by the *Control* component. This component supervises the active conditions and actions from the *Context Monitoring* and *Adaptation* components. The connections between the *Control*, *Action* and *Condition* component make it possible for the components to work together closely and to incorporate and connect the *Context Monitoring* and *Adaptation* component. The *Adaptation* component on the bottom left of Figure 1 is responsible for the execution of actions in response to a captured context change. The *Adaptation Features* component consists of several subcomponents (like the *AudioOutAction* or the *FaceUserAction* in Figure 1) specified to each execute a respective adaptation. To do so, the subcomponents make use of different parts of the *System API* to access the needed functions. Some of these APIs are for example for *AR API*, which is necessary to influence the AR part of the application, or the *Native API*, which is used to influence native system functions like the application or system language.

To achieve context-awareness, it is necessary to connect adaptations to the context changes that should trigger them. In AARCon, this is done by adding one or more conditions to an action, which will react to changes in the context features monitored by the conditions with the adaptation which is implemented for it. To be more specific, an action will become active and get executed as soon as all of its conditions are fulfilled at the same time. For cleanup purposes, there is also an *unexecute* method, which will be called when one or more of an action’s conditions are not fulfilled anymore (making the action inactive) and which is supposed to be used to reverse any effects of the action’s execution that should only be active as long as its conditions are true.

The constant monitoring of the context through the conditions and the prompt execution of the adaptations through the action is ensured by the control component. The control component acts as an observer for all of the condition and action components. Conditions report to the control when they detect a change in the context feature which they monitor, therefore they report either on it newly being true or newly being false. For example, if the camera suddenly picks up very little light, the *DarkCondition* turns true and reports this to the control. Actions report a change to the control when they are either executed (thereby executing an adaptation) or unexecuted (reversing an adaptation). As the actions depend on their registered conditions for changing, they report to the control if either all their conditions are true and were not all true before (action gets executed), or if all conditions were true before and at least one newly turned false (action gets unexecuted).

To make sure the context changes are detected, the control component regularly updates itself. If the update method is called, each of the registered actions is evaluated in order of their priority value and in turn evaluates its respective conditions. The conditions then check whether a context change has happened since the last evaluation and return the new state of the context to the calling action. If a change occurred, the condition also notifies the observing control component, causing a new update process. The actions receive the result of each of their respective conditions and react accordingly (for example by being calling their execute method). If an action is executed or unexecuted, it notifies the observing control component, so a new update will be executed in case any context features were impacted by the action.

3 IMPLEMENTATION

To achieve context-aware adaptation for AR, the general structure of the framework described previously had to be implemented. The implementation, which is available on

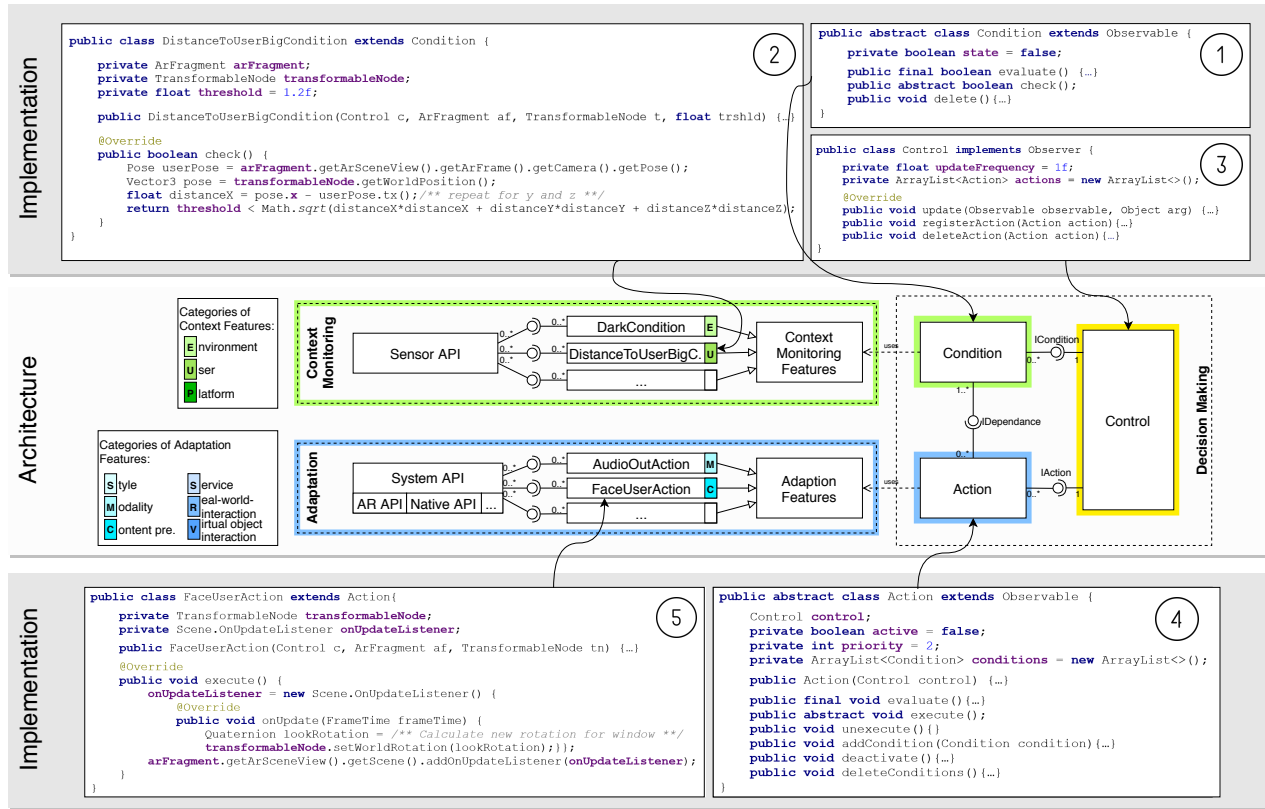


Figure 1: The main components and their implementation in AARCon

GitHub³, was done for Android's ARCore⁴, which offers markerless AR for selected Android smartphones. Mobile handheld AR on Android was chosen because, firstly, AR on smartphones is much more accessible to a wide range of people, as they do not need a special, expensive device (like a headset) to use it. Secondly, using AR on a handheld smartphone lets the users keep on using their phone the way they are used to it (handheld) and also eliminates the need for a headgear to hold the smartphone. Thirdly, the choice of Android as an operating system reaches the biggest user base, as Android has the biggest market share in smartphone OS [8] and offers a good foundation to easily start programming AR apps by ARCore with its detailed documentation and the Sceneform library, which helps to take care of the fundamental challenges in AR apps.

Context and Adaption Features

As the monitored context, as well as the possible adaptations are very diverse, AARCon proposes a taxonomy which categorizes context and adaptations to offer a more structured approach.

The context observed by the *Condition* subclasses is categorized into *Environment*, *User*, and *Platform Context*. The

Environment Context includes everything that impacts the system from the outside, such as noise or objects in the real world. *User Context* denotes any information available about the user, such as age or experience, but also the social context the user is currently in. Any information about the platform on which the system is running, like the screen size or the compatibility with different software kits, is summarized in the *Platform Context*.

The adaptation features, which are executed by the *Action* subclasses, are divided into the *Style Changes*, *Modality Changes*, *Service Changes*, *Content Presentation Changes*, *Real-World- and Virtual-Object-Interaction Changes*. The *Style Features* change the look or behaviour of single elements, the *Modality Features* treat the sensory input the user utilizes to interact with the application and *Content Presentation* features treat the way contents are presented to the user on the screen. Furthermore, the *Service Change Features* describe changes made on the device level regarding the type of device or features it offers, the *Real-World-Interaction* treats actions that are tied to objects from the real, physical surroundings of the device and the *Virtual Object Interaction Features* describe *Adaption Features* about interacting with virtual objects in an AR scene.

³ <https://github.com/S-Krings/AARCon>

⁴ <https://developers.google.com/ar/discover>

Extension of Actions and Conditions

The *Decision Making* component, as shown in Figure 1, has three sub-components: The *Conditions*, which are responsible for monitoring the context features, the *Actions*, which contain the code for the adaptations and the *Control*, which makes the other components work together.

Each of these sub-components is implemented in an own class. But while the control class (③ in Figure 1) is implemented ready to use (it can just be instantiated and used), the condition and the action main classes are abstract classes and therefore have to be extended and implemented. The cause for this construct is the fact that the context monitoring and adaptation features, which will be represented by the *Condition* and *Action* (sub-)classes, are so unique that each one needs their own code to serve their purpose. To ensure this, while also keeping the programming effort minimal, the logic which is needed in all conditions or actions for the general framework functions is preimplemented in the abstract condition (①) and action (④) class respectively, while the feature specific code is implemented in the subclasses (e.g. ② and ⑤).

As most of the functions that a user needs to extend their code with AARCon to introduce context awareness are specific to the context features and adaptation features, they would have to implement many condition and action subclasses. To reduce user effort, AARCon contains a broad range of preimplemented conditions and actions that extend the condition and action classes the same way a user would have to, giving users an example on how to implement own conditions and actions and offering a selection of ready-to-use classes (see Figure 2).

Examples for the classes' effects can be seen in the screenshots in section 4, e.g. *UsedSeldomCondition* and *ChangeControlRlementsAction* in Figure 3, *TrueCondition* and *ChangePoseToUserAction* in Figure 4, *DarkCondition* and *ChangeToVoiceInterfaceInAction* in Figure 5 and *DistanceToUserBigCondition* and *ChangeDetailAction* in Figure 6.

To keep AARCon organized, the classes are structured into several packages. The condition class and its (currently) 10 subclasses are collected in the *Conditions* package, while the more than 15 available action subclasses are located in the *Actions* package. The control class is located in the main AARCon package, as it is not supposed to be extended, but to stand for itself. As mentioned above, the condition- and action classes have to be extended to enable AARCon to do context monitoring and adaptation. To make the extension easy, the only methods that have to be implemented in the extension classes are the check method on the condition, which should be filled with code returning whether the context feature of interest fits the criteria that is to be monitored and, respectively, the execute method in the action, which

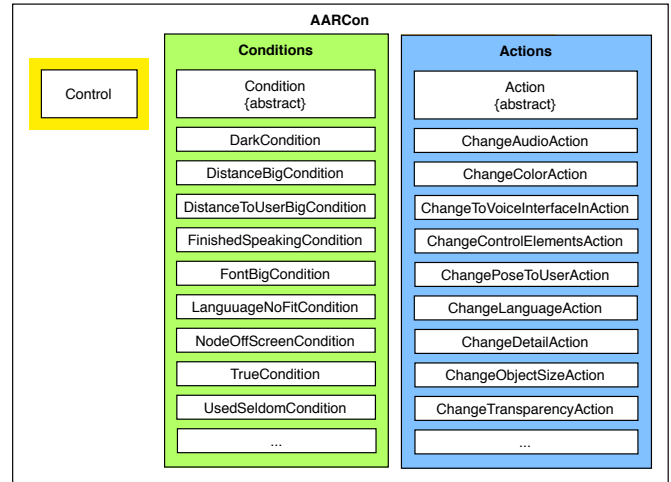


Figure 2: Overview of the preimplemented Conditions and Actions in AARCon

should be implemented to execute the desired adaptation. In the action class, the unexecute method can optionally be implemented too in case any changes of the execute method should be reversed or cleaned up.

4 CASE STUDY - PRINTER MAINTENANCE APPLICATION

As a case study, a context-aware AR app for printer maintenance was developed using AARCon. The app guides its user through the process of exchanging the ink cartridges of a HP Photosmart 7520 printer⁵ step by step, with each step being described in a text window and illustrated by 3D arrows and other elements arranged on the printer. Using AARCon, the application was made context-aware. Some of the context-aware adaptations are illustrated with screenshots in Figure 3 to Figure 6, with the left image illustrating the state before the adaptation and the right image showing it after adapting.

In Figure 3, the effect of an adaptation responding to the user's experience is shown. It displays example control elements to illustrate how to do different transformations on 3D-objects and thereby shows the user how they can, for example, rotate objects (Figure 3). As the action responsible for displaying the illustrations is connected to a condition monitoring the number of app uses, it is only executed on the first five uses of the app, so the user can use the app undisturbed once they got used to the controls.

While working on the printer, the user has to access the printer from several angles. This can cause them to look at the message with the instructions from a very steep angle, which makes it hard or impossible to read. Of course, the user could move away from the printer, read the message and then go back to printer again, but that would be rather

⁵ <https://cutt.ly/FthCIQ4>



Figure 3: Instructions on how to turn elements are only shown to new users.



Figure 5: The application switches to a voice interface if the camera is covered. (Spoken texts were added to screenshot)

inconvenient and disruptive for the workflow. For this reason, the application uses an action which rotates the specified object, in this case, the message window, towards the user at all times (see Figure 4) to make sure it is always readable.

There is also a condition which turns true whenever the camera receives very little light input, which is usually the case when the device was laid down, for example if the user needs their hands free. As this prevents the user from seeing any objects of the AR application, a voice interface is activated (see Figure 5). It makes sure the description of the current task is read out to the user and it enables them to interact with the application using voice commands. This allows the user to interact with the application even if they are currently not able to hold the device to use it in AR mode.

Figure 6 shows a change in the instruction window's level of detail. This is achieved using the action for this in combination with a condition which reacts to the user's distance to the printer, so the detail is lowered if the user is more than for example 1.2 meters away from the printer. Adjusting the level of detail can help the user to focus on the currently important task. It can also be used to make the user come closer to important objects or to create a simpler and tidier AR environment by removing information that is currently unnecessary.

These features, amongst others, aim to enhance the user's experience using the printer maintenance application. As



Figure 4: The window is rotating to always face the user regardless of their position.



Figure 6: When the user moves away from the printer, the level of detail in the text on the window is decreased.

they are all executed automatically based on context information, the user does not have to do anything to get an application which is at all times customized to the current situation.

5 RELATED WORK

Context-awareness has been researched before in several works, such as the AdaptUI framework [12], which treats context-awareness in web applications, or [10], which focuses on context-aware virtual reality. There are also component-based approaches on adaptive UIs [11]. All these works are already focusing on context-awareness, but are not targeting the field of context-aware AR. In the following, we will evaluate several existing approaches on context-awareness in AR and compare them to our AARCon framework based on several functional and non-functional requirements. On a functional level, the approaches are compared based on whether they consider the aforementioned context and adaptation features and whether they provide a tool-support for the development of context-aware AR apps. The non-functional requirements cover whether an approach is modular, extensible to new use cases and reusable.

CAMAR (short for context-aware Mobile Augmented Reality) [7] is an approach supporting context-awareness in mobile AR focusing on the user context. It enables the user to customize the presentation of virtual content and to share this information with other users selectively, depending on

Legend	Functional Requirements										Non-Functional Requirements				Abbreviations:			
	CF			AF							T	M	E	R	CF	Context Features	CP	Content Presentation
	U	E	P	Sty.	M	CP	Ser.	RWI	VOI	AF					Adaptation Features	Ser.	Service	
										U					User	RWI	Real-World-Interaction	
+ Completely Fulfills																		
o Partly fulfills																		
- Does Not Fulfill																		
CAMAR [6]	+	o	-	-	-	+	-	+	+	+	+	+	+	+				
Computational MR [5]	+	+	+	+	+	+	+	-	+	o	+	+	+	+	o			
PAR [4]	+	+	+	+	+	+	+	+	+	o	-	+	+	+	+			
ACARS [12]	+	+	+	+	-	+	+	+	+	-	+	+	+	+	+			
AARCon	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			

Sty.	Style	M	Modularity
E	Extensibility		
R	Reusability		

Figure 7: Summary of the related approaches

the context. Furthermore, it includes the development framework UCAM. The approach mainly focuses on user context, but provides a modular and extensible structure and the UCAM tooling.

The work by Lindlbauer et al. [6] focuses on context-aware adaptation of interfaces in Mixed Reality (MR), with the main adaptation points being what content is displayed and where and how much detail of it is shown, depending on the user's tasks and cognitive load. In terms of context features, it is clear that Lindlbauer's framework focuses on the user, but in contrast to CAMAR, it also takes context features from the environment and platform category into account. Furthermore, it covers most of the required adaptation features, leaving only the service and the virtual object interaction partly out of consideration. The approach also offers tooling in form of an open source implementation and is easily extensible in its planned use.

In the paper by Grubert et al. [5], the concept of Pervasive Augmented Reality (PAR) is introduced as a future development in AR. A taxonomy for PAR and context-aware AR that classifies context sources and targets is presented. The context sources correspond very closely to AARCon's ones and the adaptation features are also mostly well comparable. A tooling, however, is not provided, but in its theoretical form, the approach is extensible and reusable due to its modular structure.

The ACARS approach by Zhu et al. [13] focuses on context-aware AR in maintenance scenarios. It aims to offer an adapting interface which shows maintenance instructions which can be added and changed by the user on-site and via a desktop interface. ACARS takes a big range of context, but focuses its adaptations on the presentation of information, leaving for example modality changes untouched. Tooling is provided as well as a modular, extensible framework.

Figure 7 illustrates that only the AARCon framework fulfills the defined requirements. The approaches either do not include an adequate selection of adaptations (like [6]), or they are theoretical and do not offer any kind of architecture or framework to make them usable in practice (see [5]). Also, they are often tied to a specific task and are specialized for it, so extending them to expand their abilities is only possible in a very limited way. AARCon advances the state-of-the-art by combining a broad range of adaptations with a practical approach, which is not tied to a specific task.

6 CONCLUSION AND OUTLOOK

In conclusion, AARCon offers a framework for developing context-aware AR applications developed in a modular approach. Due to this it is highly extensible to different use cases and offers a broad range of functions. We have shown the practical benefit of AARCon based on a context-aware AR printer maintenance application. In ongoing work, we are analyzing the efficiency and effectiveness of the AARCon framework from developer's perspective. Furthermore, we will conduct usability tests to investigate the usability of the resulting context-aware AR applications based on AARCon.

REFERENCES

- [1] G.D. Abowd and A.K. Dey. 1999. Towards a Better Understanding of Context and Context-Awareness.
- [2] Ronald T. Azuma. 1997. A Survey of Augmented Reality. *Presence: Teleoper. Virtual Environ.* 6, 4 (Aug. 1997), 355–385.
- [3] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* (2003).
- [4] T.P. Caudell. 1992. Augmented reality: An application of heads-up display technology to manual manufacturing processes.
- [5] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2016. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23 (03 2016), 1–1.
- [6] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *UIST*.
- [7] Sejin Oh, Woontack Woo, et al. 2009. CAMAR: Context-aware mobile augmented reality in smart space. *Proc. of IWUVR* 9 (2009), 48–51.
- [8] Statcounter. 2019. Mobile Operating System Market Share Worldwide.
- [9] Ivan E. Sutherland. 1968. A Head-Mounted Three Dimensional Display. In *Fall Joint Computer Conference*. Association for Computing Machinery, New York, NY, USA, 757–764.
- [10] Enes Yigitbas, Joshua Heindörfer, and Gregor Engels. 2019. A Context-aware Virtual Reality First Aid Training Application. In *Mensch und Computer 2019*. GI / ACM, 885–888.
- [11] Enes Yigitbas, Klementina Josifovska, Ivan Jovanovikj, Ferhat Kalinci, Anthony Anjorin, and Gregor Engels. 2019. Component-based development of adaptive user interfaces. In *Symposium on Engineering Interactive Computing Systems, EICS 2019*. ACM, 13:1–13:7.
- [12] Enes Yigitbas, Stefan Sauer, and Gregor Engels. 2017. Adapt-UI: an IDE supporting model-driven development of self-adaptive UIs. In *Symposium on Engineering Interactive Computing Systems, EICS 2017*. ACM, 99–104.
- [13] J. Zhu, S. K. Ong, and A. Y. C. Nee. 2013. An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology* 66, 9 (2013), 1699–1714.

Meta-Level Support for Facilitating Participation in Website (Re-)Design Activities

Henrik Hertel

University of Rostock, Germany
henrik.hertel@uni-rostock.de

Anke Dittmar

University of Rostock, Germany
anke.dittmar@uni-rostock.de

Doritt Linke

University of Rostock, Germany
doritt.linke@uni-rostock.de

ABSTRACT

Complex websites need to evolve continuously as they contribute to shaping the relationships between involved stakeholders. The paper presents conceptual ideas and a prototypical tool to facilitate participation in web redesign activities. The proposed meta-level support allows to augment dedicated parts of web pages by sets of design options. End users are provided with means to explore and assess the alternative design ideas in a parallel prototyping manner and to create and share annotations. A real-world example is used to motivate and illustrate the suggested approach. The contribution to end-user design and cultures of participation is discussed.

CCS CONCEPTS

• **Human-centered computing** → *User centered design; Participatory design; Interface design prototyping; Web-based interaction.*

KEYWORDS

Meta-design; parallel prototyping; web-augmentation

ACM Reference Format:

Henrik Hertel, Anke Dittmar, and Doritt Linke. 2020. Meta-Level Support for Facilitating Participation in Website (Re-)Design Activities. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398641>

1 INTRODUCTION

The evolution of web applications from static, content-based web pages to dynamic applications with functionality and services is accompanied by increased user participation. While

in the Web 1.0-era, users were mainly limited to reading and form filling, Web 2.0 applications enabled them to switch between the role of consumer and (collaborative) content producer. By blogging or wiki editing users become ‘prosumers’ and Olsina et al. [15] describe corresponding websites as being in an endless beta condition. Mashups and web augmentation even allow web users to compose new applications from existing web resources and to customize websites, respectively [3]. Such means for end-user development support the idea of ‘open systems’ that users can modify and evolve at use time but also face challenges such as participation overload [8]. Depending on their expertise and motivation, individuals can or want to contribute to a website at different levels (e.g., as local developer, content provider, or just as reader). Hence, in their meta-design framework, Fischer et al. [7, 8] call for a variety of technological environments to support a broad spectrum of participation scenarios.

This paper investigates the idea of end-user design environments for web applications with a focus on technical aspects. In particular, we look at websites with diverse stakeholders where participation in evolutionary redesign activities is needed to adapt them to the changing needs and constraints. Design requires both elaboration of meaningfully distinct design options and making choices [2]. The paper proposes meta-level tool support which allows (local) developers to augment dedicated parts of websites by sets of design options. End users can choose between and comment on those options. Additionally, the creation and sharing of annotations is supported.

The paper is organized as follows. Section 2 introduces a motivating example that is also used to illustrate the conceptual design, prototypical implementation, and intended use of the presented tool support. Related work, contributions to end-user design, and future work are discussed in section 3. We finish with some conclusions in section 4.

2 META-LEVEL TOOL SUPPORT

Like any other artifact, websites are shaped by, and in turn shape, the social and organizational contexts of involved stakeholders. This section presents tool support (based on web augmentation techniques) aimed at facilitating participation in redesigning web applications with diverse stakeholders (e.g., institutional websites). In such cases, stakeholders

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06...\$15.00

<https://doi.org/10.1145/3393672.3398641>

do not form a ‘monolithic community’ and the complexity of the application results from different interests in presenting and using information. We first look at a small example from our own experience: the website of a university department.

Motivating Example

Websites of university departments are often created and managed by content management systems and need to fit into the university website. User groups include prospective students, undergraduate and graduate students (local, national, international), visitors, faculty/staff (department, university, other universities), parents, research consumers, research program managers, and press [16]. Ritter et al. show in their task analysis that “nearly all the constituencies of a university department now interact with its Web site, and that nearly all the tasks and work that departments do are mirrored on the Web site as well” [16]. The authors admit that the range of users and the breadth of tasks “suggests a wider and deeper use than we thought we would see” [16].

Two authors of this paper observed a similar underestimation of the complexity of their department website when they were involved in a redesign process triggered by a new university web layout. For instance, the department management initially assumed an automated ‘content transfer’ from the existing to the new layout. Most parts of the website were completely revised. The ‘study & teaching’ area was redesigned together with graduate students participating in an interaction design course, with three teacher representatives, and with participants in a design workshop. The new design took more account of the needs of prospective and beginner students. The revision of the ‘research’ area was carried out by a small group of professors, discussed among professors in one of their regular meetings, and led to minor modifications. Students and staff members were asked to comment on the resulting online prototype. Some of the new design ideas could only be implemented with their voluntary contributions. It was generally more effective to directly contacting colleagues and students (face-to-face and email) than to sending out group mails. Not surprisingly, many participants of the above mentioned meetings and workshop later provided contributions.

Complex websites evolve over a sustained span of time. In our understanding, their redesign does not refer to required periodic updates but to more comprehensive revisions with exploratory steps (e.g., due to strategic refocusing, changes in user expectations or new design constraints). Besides general challenges of user participation (e.g., securing time and space to get committed in design processes, lack of design skills such as brainstorming, change of perspectives, concept creation [13]), we also noticed limitations of existing prototyping approaches to engage a broader audience in co-design activities.

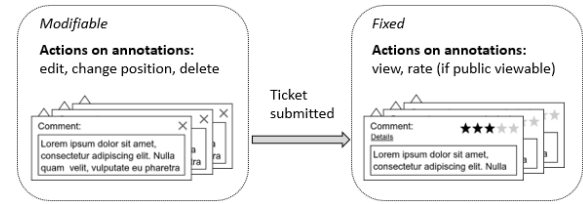


Figure 1: Annotations are bundled together in tickets.

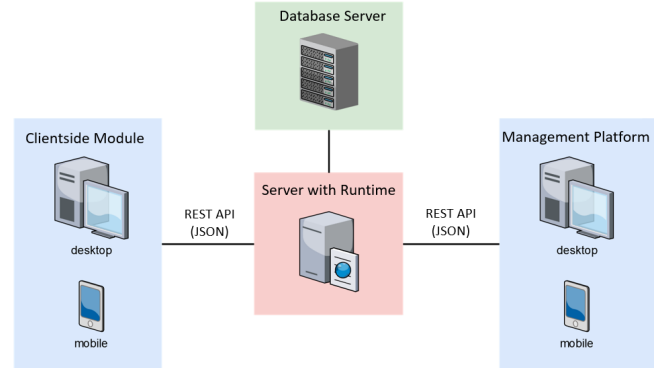


Figure 2: Main components of the end-user design environment.

Conceptual Design and Architecture

In this work, we employ ideas from parallel prototyping and collaborative annotations to contribute to end-user design environments that facilitate different levels of participation in the addressed domain. Such environments need to support an alternation between phases where new ideas for parts of the website under consideration are developed (idea generation) and phases where those ideas are compared and assessed (idea convergence). The proposed tool is especially aimed at supporting participation in idea convergence by sharing and discussing sets of design options. (This, of course, can prepare idea generation again).

We refer to our tool support as ‘meta-level’ support because web augmentation techniques are used to easily integrate ‘throw-away’ implementations of design options and annotation mechanisms into running web applications. Web augmentation builds on top of the rendering of an existing website [3]. In our case, alternative designs for particular parts of web pages are added and can be explored in a parallel prototyping manner. According to Dow et al. [4], parallel prototyping, that is the creation and sharing of multiple designs foster exploration, prevent people from framing design problems too narrowly, and improve results [4]. It also helps designers to react appropriately to critique and feedback on design ideas [4, 5]. Our augmented web pages offer the following features to end users.

Exploration and choice:

- Switching between and viewing design options for dedicated parts of web pages.
- Choosing between options, commenting, and posting.

Collaborative referencing of web page elements:

- Creation, editing, and positioning of annotations,
- Posting of tickets (optionally as public viewable),
- Rating of public viewable annotations.

Figure 1 illustrates the ‘life cycle’ of annotations, bundled together in tickets. As long as a ticket is not submitted the annotations can be modified. If it is submitted as ‘public viewable’ the annotations can be viewed and rated by all users of the augmented web pages. Otherwise, they can only be viewed by the creator or via the management platform.

Figure 2 shows the overall architecture of the meta-level tool support consisting of three main components. The *back-end* offers a Representational State Transfer (REST) API to communicate with the client-side modules for end users and the client-side management platform. The *management platform* (to be implemented as Single Page Application, SPA) can be used by (local) designers and developers to augment their website and make it accessible to all end users or to sub-groups. For each website domain, after being registered and verified, projects can be created and managed which span the web pages under redesign. In the prototypical implementation described below HTML sections and a CSS preprocessor are used to specify and handle alternative designs. It should be noted that design options can and perhaps even should be ‘sketchy’ to convey the basic ideas (the second scenario in the next sub-section illustrates this). A project aggregates all tickets and selections of favored design options that are submitted for the assigned web pages. Feedback information can be viewed and shared by email, tickets can be closed for rating or deleted etc.

The *client-side module* adds a meta-level feedback area with the above described functionality to web pages under consideration. If alternative designs are specified for some parts of a web page the feedback area provides local dialog boxes that allow end users to switch between the different designs, to select one option and give comments.

Prototype and Scenarios of Use

The prototypical implementation of the proposed tool support employs widely used web technologies and development tools including the following ones.

- *Client-side module*: TypeScript and tslint¹ for static code analysis, HTML5, SCSS and node-sass² as CSS preprocessor, webpack³ as module bundler.

¹<https://palantir.github.io/tslint>

²<https://github.com/sass/node-sass>

³<https://webpack.js.org>

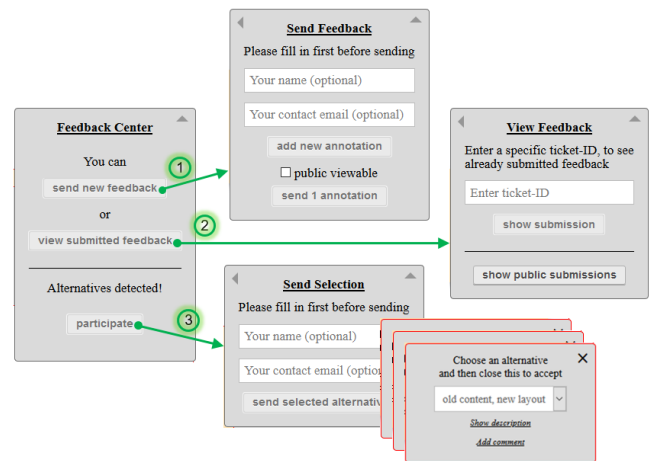


Figure 3: User interface of the meta-level feedback area.

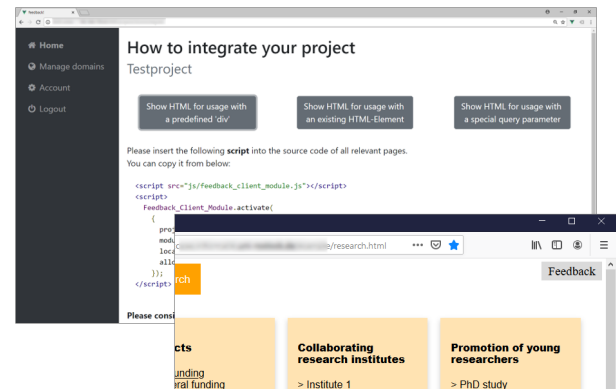


Figure 4: Dianne injects the feedback area to web pages (see button labeled ‘Feedback’ in the bottom window).

- *Management platform (SPA)*: JavaScript, HTML5 and CSS, Vue.js as JavaScript framework for SPAs, bootstrap⁴ as CSS framework, webpack, auth0.js⁵ (library for SPAs to do authentication), SignalR⁶ (library for bidirectional communication with server-side module).
- *Backend*: C#, ASP.NET Core⁷ as cross-platform web-development framework, Microsoft SQL Server LocalDB (database management system), EF Core⁸ as cross-platform object-relational mapping framework, SignalR, Fluent-Email⁹ (library for sending emails).

Figure 3 depicts the user interface of the feedback area that augments web pages: (1) submission of tickets, (2) viewing

⁴<https://getbootstrap.com>

⁵<https://auth0.com/docs/libraries/auth0js/v9>

⁶<https://www.asp.net/signalr>

⁷<https://github.com/aspnet/Home>

⁸<https://github.com/aspnet/EntityFrameworkCore>

⁹<https://github.com/lukenode/FluentEmail>

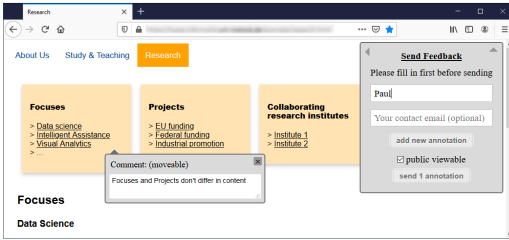


Figure 5: Paul creates an annotation.

of public and private annotations, (3) exploration of design options for one or more parts of a web page. The relevant elements of the prototype are now demonstrated by two simple scenarios of use where we take up the example described at the beginning of this section. Let us imagine that the ‘research’ area of the website is under redesign and the following actors from the department are involved:

- *end-user designers*:
Paul (student), Claire (PhD student), Ann (secretary),
- *local developer and content manager*:
Dianne (system engineer),
- *local designer*: Carl (research associate).

Scenario I. Dianne and Carl want to initiate a discussion about the research area. Dianne logs in to the management platform and adds a new project to the website domain. The tool offers three modes to embed the feedback area.

- *Mode 1*: a button is automatically embedded in the upper right corner of the web page to activate the feedback area (by using a predefined HTML `<div>` tag).
- *Mode 2*: the activation of the feedback area is attached to an existing element of the web page.
- *Mode 3*: similar to mode 1 but the feedback area is only included if the URL is extended by a generated query parameter.

Dianne selects mode 1 to address all users. She is asked to insert the generated script (see figure 4) into the source code of all relevant web pages to map them to the project. Figure 5 depicts Paul’s actions. He activated the feedback center and moved an annotation to the first two cards saying that they do not much differ in content. He has no more comments and submits his ticket as public viewable. When Carl enters the management platform he notices the arrival of two tickets, one sent by Paul and the other anonymously. Ann can view the annotations and ratings of these tickets as well (figure 6).

Scenario II. In the second scenario, Carl and Dianne, together with other colleagues, developed some ideas for presenting the department’s research activities. Carl asks Dianne to implement the two design options he created for the main research page and to keep it sketchy. In our prototypical implementation, parts with alternatives have to be

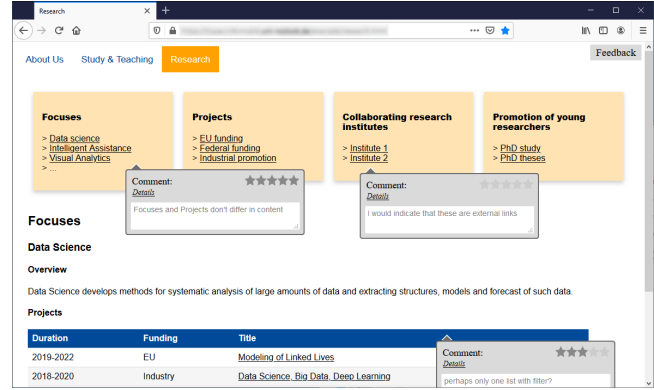


Figure 6: Ann views the annotations and ratings of public viewable tickets.

specified as HTML sections with two attributes (a unique name and a problem description) and with sub-sections representing the design options. Each design option has a title (as HTML attribute) and can again contain nested options. Additionally, SCSS code has to be specified for the options. Only one option is visible at a time, the others are hidden by the CSS declaration “display:none”. Figure 8 shows code fragments of the example in figure 7.

Dianne uses mode 3 (described above) to augment the web pages by the feedback area. She sends a group email with the generated link to the research staff members. Figure 7 shows someone switching between the design options and commenting on their choice. The option on the left-hand side is similar to the old page but with the new required layout, the option on the right-hand side adds content, there are changes in the navigation structure and presentation. Finally, Dianne and Carl view and share (by email) information about submitted choices. For example, Claire prefers the second option because research is not only presented in terms of projects (figure 9).

3 DISCUSSION, RELATED AND FUTURE WORK

Fischer et al. [8] distinguish between closed systems and open systems to discuss different assumptions about development processes and the role of the designer and user. “Closed systems typically create a sharp separation between design and use” [8] and result in interactive tools that are ‘designed for use before use’. In contrast, open systems acknowledge the co-evolution between users and (technological) systems. The aim is not to develop ‘complete’ systems but to consider design as open and continuous process. Users are included as co-designers throughout the entire existence of the system; they are provided with tools for ‘designing in use’. The approach in this paper supports the idea of open systems.

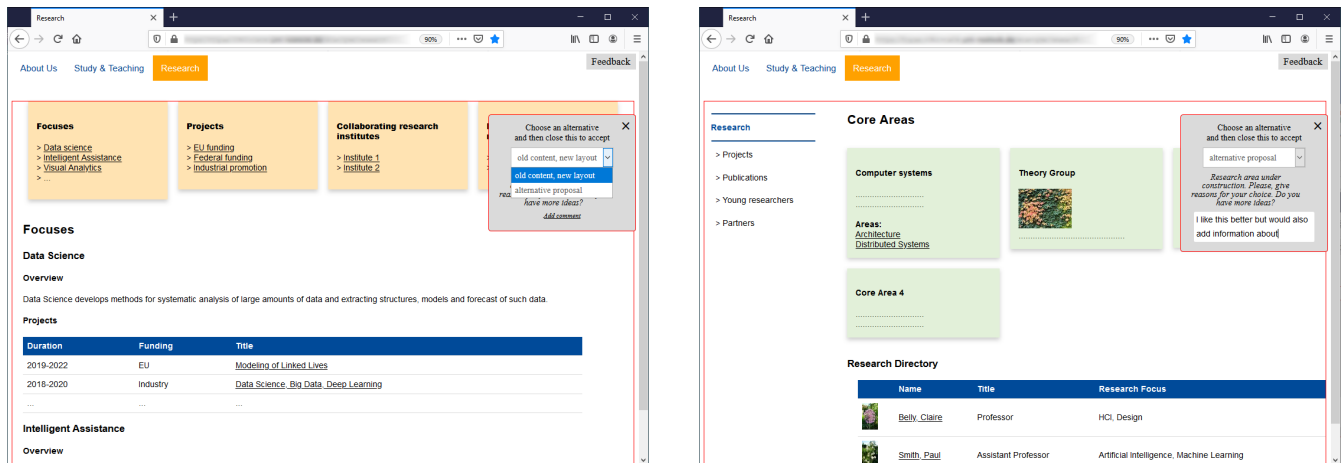


Figure 7: At the meta-level, parts with design options are framed in red, with an attached dialog box for exploration, choice and comments. Take note that the design options in the example are implemented in a sketchy way.

```

1 <section data-fm-area-name="Research"
2   data-fm-area-description="Research area under
   construction. Please,...">
3 <!-- 1. Alternative -->
4 <section data-fm-area-item-title="old content, new
   layout">
5   <div class="row">
6     <div class="column">
7       <div class="card">
8         <h4><b>Focuses</b></h4>
9       ...
10    </div>
11  </section>
12 <!-- 2. Alternative -->
13 <section data-fm-area-item-title="alternative
   proposal">
14   <div class="wrapper">
15     <div class="box1">
16     ...
17   </div>
18 </section>

```

```

1 [data-fm-area-name="Research"] {
2   /* 1. Alternative */
3   &>[data-fm-area-item-title]:nth-child(1) { ...
4   }
5   /* 2. Alternative */
6   &>[data-fm-area-item-title]:nth-child(2) {
7     display: none;
8     .vertical-menu a {
9       display: block; ...
10  }
11 }

```

Figure 8: Specification of two design options for a part of the page named “Research” (top: HTML code fragment, bottom: SCSS code fragment).

Of course, the boundary between closed and open systems is blurred. However, web development tools such as wireframe tools for early design, content management systems, and even wikis - although supporting collaboration, content-providing, and user feedback - rather assume closed systems.

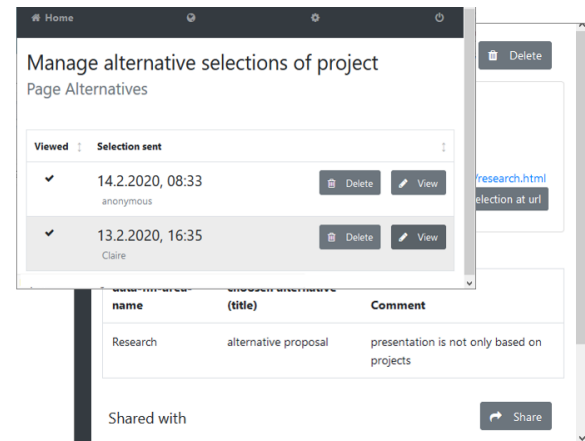


Figure 9: Carl and Dianne view feedback on design options via the management platform.

They provide less suitable means for end-user redesign which not only concerns modifications of the content but also of the behavior and presentation of websites [1]. Open systems are especially promoted by end-user development and underlying conceptual frameworks (e.g., meta-design [8]). There are currently two main categories of tools for end-user web development (apart from tools to directly create websites) [1]. Mashup tools enable end users to create new web applications by combining existing resources and services. Web augmentation tools help to customize and extend existing web pages. CrowdAdapt [14], as an example, provides a visual design environment that augments web pages to adapt their layout. Additionally, crowdsourcing is used for sharing the adaptations. Aldalur et al. [1] give a good overview of existing end-user development tools along five dimensions

(programming paradigm, subject of adaptation are two of them).

Our approach focuses less on customization and end-user programming and more on integrating exploratory design activities into website evolution. In the context of the object-oriented development paradigm, such an integration is considered in [11] and a software architecture is suggested that allows to easily compose alternative ‘throw-away extensions’ with the evolutionary prototype at run time. We use web augmentation techniques to enable a similar exploration of alternative design ideas. The proposed tool support considers the central meta-design concepts mentioned in [12]: convivial tools, domain-orientation, open, evolvable systems, underdesigned systems, and collaborative work practices. In particular, the demonstrated parallel prototyping for dedicated web page areas is a novel contribution to underdesign, a technique that provides design spaces for users and helps them to understand an existing system as incomplete and evolving.

In meta-design, “not every participant must contribute, but all participants must have opportunities to contribute when they want to” [7]. Various role systems are needed to avoid participation and collaboration overload in specific domains and situations. For instance, in the study by Fogli and Provenza [9], administrative employees become end-user developers of e-government services while citizens stay exclusively in the role of end users of these services. The negotiation of the relationships between website owners, developers of external scripts, and end users in web augmentation contexts is investigated in [6]. Fischer et al. [8] recommend to have meta-designers who technically enable and support users’ participation and maieuta designers who encourage and sustain certain ‘cultures of participation’. Websites which are in the focus of this paper have to meet the needs and interests of diverse stakeholders and different levels of engagement can be expected. As the above scenarios indicate, the suggested meta-level tool support allows to create different projects for a website in order to provide specific augmentations (sets of design options) to specific groups of end users. However, it is beyond the scope of this paper to empirically investigate experiences of maieuta designers and others in using the tool. At this stage, we have only collected initial feedback from colleagues and students. While studies on parallel prototyping [4, 5] and collaborative annotations [10] suggest that these techniques support participation, we have to carry out an in-depth experimentation to investigate the effectiveness of our tool in encouraging co-design activities. Related questions concern, for example, the number of design options which should ideally be offered to end users and whether alternative ways of representing design options have an influence on the quality of the comments.

4 CONCLUSIONS

It is increasingly accepted that richer ecologies of participation are crucial in (re-)design activities [7]. The presented tool support contributes to an end-user (re-)design environment for web applications. We have shown that it facilitates different levels of participation. Possible ways of using the tool have to be investigated in future empirical studies.

REFERENCES

- [1] I. Aldalur, M. Winckler, O. Díaz, and P. Palanque. 2017. Web Augmentation as a Promising Technology for End User Development. In *New Perspectives in End-User Development*. Springer, 433–459.
- [2] B. Buxton. 2007. *Sketching User Experiences : Getting the design right and the right design*. M. Kaufmann.
- [3] O. Díaz, C. Arellano, and M. Azanza. 2013. A Language for End-User Web Augmentation: Caring for Producers and Consumers Alike. *ACM Trans. Web* 7, 2 (2013), 51.
- [4] S. Dow, J. Fortuna, D. Schwartz, B. Altringer, D. Schwartz, and S. Klemmer. 2011. Prototyping Dynamics: Sharing Multiple Designs Improves Exploration, Group Rapport, and Results. In *Proc. of CHI (CHI '11)*. ACM, 2807–2816.
- [5] S.P. Dow, A. Glassco, J. Kass, M. Schwarz, D.L. Schwartz, and S.R. Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Comput.-Hum. Interact.* 17, 4 (2010), 18:1–18:24.
- [6] D. Firmenich, S. Firmenich, G. Rossi, M. Winckler, and D. Distant. 2015. User Interface Adaptation Using Web Augmentation Techniques: Towards a Negotiated Approach. In *Proc. of ICWE 2015 (LNCS)*, Vol. 9114. Springer, 147–164.
- [7] G. Fischer. 2011. Understanding, Fostering, and Supporting Cultures of Participation. *Interactions* 18, 3 (2011), 42–53.
- [8] G. Fischer, D. Fogli, and A. Piccinno. 2017. Revisiting and Broadening the Meta-Design Framework for End-User Development. In *New Perspectives in End-User Development*. Springer, 61–97.
- [9] D. Fogli and L.P. Provenza. 2011. End-User Development of e-Government Services through Meta-modeling. In *Proc. of IS-EUD 2011 (LNCS)*, Vol. 6654. Springer, 107–122.
- [10] F. Gao. 2013. A case study of using a social annotation tool to support collaboratively learning. *The Internet and Higher Education* 17, 4 (2013), 76–83.
- [11] H. Hertel and A. Dittmar. 2017. Design support for integrated evolutionary and exploratory prototyping. In *Proc. of EICS*. ACM, 105–110.
- [12] B. Koehne, D. F. Redmiles, and G. Fischer. 2011. Extending the Meta-design Theory: Engaging Participants as Active Contributors in Virtual Worlds. In *Proc. of IS-EUD 2011 (LNCS)*, Vol. 6654. Springer, 264–269.
- [13] P. Marti and L. Bannon. 2009. Exploring User-Centred Design in Practice: Some Caveats. *Knowledge, Technology & Policy* 22 (2009), 7–15.
- [14] M. Nebeling, M. Speicher, and M.C. Norrie. 2013. CrowdAdapt: enabling crowdsourced web page adaptation for individual viewing conditions and preferences.. In *Proc. of EICS 2013*. ACM, 23–32.
- [15] L. Olsina, R. Sassano, and L. Mich. 2008. Specifying Quality Requirements for the Web 2.0 Applications. In *Proc. of the 7th International Workshop on Web-Oriented Software Technologies (IWOST'2008)*.
- [16] F. E. Ritter, Andrew R. Freed, and Onida L. M. Haskett. 2005. Discovering user information needs: the case of university department web sites. *Interactions* 12, 5 (2005), 19–27.

Closed Loop Feedback Nudges Using Nerve Stimulation

Rohan Hundia
University of St Andrews
St Andrews, United Kingdom
rh208@st-andrews.ac.uk

Aditya Vijayvargiya
LDRP Institute of Technology
Gujarat, India
adityavijayvargiy@gmail.com

Aaron Quigley
University of St Andrews
St Andrews, United Kingdom
aquigley@st-andrews.ac.uk

ABSTRACT

Technology mediated nudges to alter and aid decision making have been explored using various modalities in HCI. In this paper we propose the development of user interfaces which make use of an electrical nerve stimulation mechanism to act as a 'closed-loop' nudging aid for navigation, enhanced nudge feedback based gaming and human to human interaction. We demonstrate three major application interfaces a) using nerve nudges incorporated with a navigation API to assist walking on a predefined route tested on people without visual impairments which could be further extended as a useful application scenario for people with visual impairments. b) Improving gaming performance using forced motor events from nerve stimulation in a bidirectional car racing game. c) enabling human to human control which could be further used in neuro-feedback mechanisms. We also discuss future applications of our nudging mechanism in accident-avoidance systems and human to human learning.

CCS CONCEPTS

• **Human-centered interaction** → **Human computer interaction**; Interaction techniques;

KEYWORDS

TENS; nerve stimulation; nudges; forced events

ACM Reference Format:

Rohan Hundia, Aditya Vijayvargiya, and Aaron Quigley. 2020. Closed Loop Feedback Nudges Using Nerve Stimulation. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3393672.3398642>

1 INTRODUCTION

Behavioural nudges are techniques used to influence or facilitate change in someone's behaviour by altering the underlying 'choice architecture' without forbidding any alternative action decisions. By contrast, 'forced' action events lead to behavioural alterations without the option for the user of having control over the underlying action.

Permission to make digital or hard copies of all or part 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7984-7/20/06.

<https://doi.org/10.1145/3393672.3398642>

Nudges have been used extensively in various fields such as economics, policy making, marketing, healthcare, human-computer interfaces etc. In policy making, organ donation 'opt-out' nudge consent policy has been adopted by various countries such as Spain and France, whereby the default option for the people of the country assumes that citizens are considered willing to be organ donors unless they opt-out which thereby increases the number of organ donors [16]. Similarly, 'up-selling' nudge techniques are often employed in restaurants where servers offer additional options with a meal to increase profits [17].

In HCI, technology mediated nudges are used extensively in behavioural change technologies and user interfaces to reduce user abandonment rates and are incorporated in design strategies to increase user engagement. Digital nudges are also used by various apps to build and reinforce positive habits and health [8]. A research study by Caraban et al. [4] proposes a framework of 23 nudging mechanisms including opt-out policies, placebos, default options, positioning, subliminal priming etc. which are used in a majority of application scenarios. For example, persuasive technologies make use of priming cues to alter choice actions [2]. One of the most commonly used techniques are visual nudges which make use of our peripheral awareness in the visual field to affect our attention system subconsciously and thereby our decision making. Such visual peripheral nudges are in the form of subliminal primes or subconscious visual cues in our peripheral vision. They are often used on food packaging as labels or designs, advertisement hoardings to increase sales and even digital interfaces in the form of side-bar pop-ups [6, 15]. Nudges can also be used in brain to brain interfaces in the form of neuro-nudges or 'forced action' neural events that are transmitted as information from the sender to the receiver and is usually associated with stimulating regions of the motor cortex of the receiver when motor related brain activity in the sender is detected [14]. Traditionally, brain to brain interfaces combine electroencephalography (EEG) to analyse brain activity and transcranial magnetic stimulation (TMS) to stimulate certain brain regions.

In our research, we explore the use of 'nudges' and 'forced events' using a lesser known modality - electrical nerve stimulation - in comparison to the more traditional nudge systems based on vibrations, sound or visual feedback. We explore useful application scenarios utilizing a closed loop trigger mechanism. Our approach makes use of transcranial electrical nerve stimulation (TENS) - Med Fit 3, a dual channel TENS machine - with three different user interfaces to effectively enforce motor nudges or 'forced events' to explore use cases as described below:

A) A Navigation app we developed using the MapBox API linked with TENS (electrodes placed either over the radial nerve of the arm, or the fibular nerve over the leg) to provide motor nudges during left or right turns during walking.

B) A feedback car racing game developed in Python that provides forced trigger events (using TENS) and elicits motor movement over certain fingers of the left or right hand to control the car at high speeds (where control becomes difficult) over route turns. We hypothesise that this forced motor event triggered gaming could improve performance in manual trials (post forced-event trials).

C) A human to human control interface using 8 channel EEG and TENS to 'force' motor activity of one person into another. This work explores the possibilities of this mechanism as a neurofeedback control to stimulate movement.

2 RELATED WORK

Prior research in HCI has explored nudges for recommender systems, in privacy-invasive aware mobile apps [1], to enforce discipline and improve productivity, and vibrating phones to reduce digital consumption [11] and in BCIs to navigate cursors on a computer screen using frequency control 'nudges' in locked in patients [10]. The use of electrical nerve stimulation has been often targeted towards pain relief systems, facilitating neurological therapy [3], with limited work being done using neuromuscular and nerve stimulation as 'forced' haptic feedback events in immersive gaming setups [9]. Traditionally, electric muscle stimulation (EMS) methods have been used extensively in HCI to co-create visual art works [5], adding forced haptic based muscle stimulation for training in virtual environments and increase performance in VR tasks [13]. Additionally, EMS based techniques have also been used to simulate haptic rendering while playing percussive instruments such as teaching rhythms while playing drums [7] and also in actuated walking navigation interfaces based on remote-control based button presses to stimulate and actuate muscles for navigation [12]. In this paper, we explore the development and integration of a TENS machine with three major user interfaces and application scenarios. Although TENS and EMS largely have the same desired stimulation output, a difference occurs in location-precision of stimulation. Stimulated movements associated with EMS are mostly targeted towards individual muscles, however TENS allows for stimulating nerve impulses in different locations along the nerve and between various nerves which could lead to greater control over desired stimulation. For example, in order to get individual finger movements of the hand we can stimulate individual nerves (radial, ulnar, medial nerve etc) and we could also control the type of movement based on where the stimulation is applied along a nerve. Additionally, we focus on building a closed-loop mechanism alongside investigating different levels of stimulation (leading to nudges or 'forced events') in varying applications.

3 DESIGN AND STATE DETECTION

All three interfaces made use of a TENS machine integrated with a micro-controller as described below:

Hardware Design - Figure 1 (B): We used the Med Fit 3 TENS machine (clinically approved), which is a dual channel system, where each channel comprises adhesive electrode pads for nerve stimulation. The machine comprises of a manual control box to adjust frequency (number of pulses per second), pulse width and pulse amplitude to varying intensity of stimulation (0-80mA). The

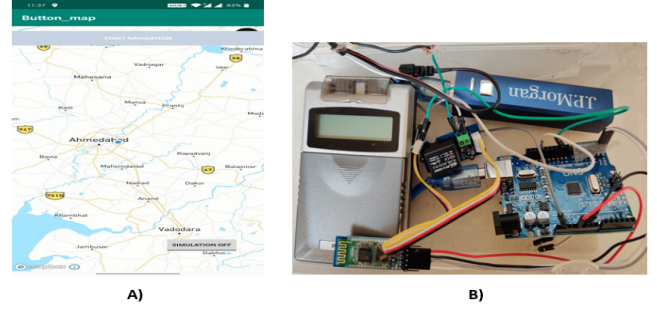


Figure 1: Design – A) Front-End App Interface for our navigation nudge interface using MapBox API, B) Hardware Design for nerve stimulation in all our applications comprising MedFit 3 TENS, Relay, Arduino Uno, HC05 and a powerbank

frequency decides the number of electric impulses applied in a second leading to nerve contractions and expansions. A higher frequency stimulation would be 'felt' as longer by a subject simply because more number of electric impulses are applied in a single second. The intensity of the stimulation is based on the amount of electric current applied (amplitude of single pulse) in amperes and therefore determines the strength of a single electric pulse. A single channel consists of two electrode connections for desired stimulation. We bifurcated each of the electrode connections within a single channel, which were connected to separate pins of the 5V normally open relay switch which was then connected to the digital output pins of the Arduino Uno micro-controller. A HC05 Bluetooth module was also connected to the Arduino. The Arduino acted as a receiver of processed data from a mobile app or a computer application and triggered the relay which further acted as a switch to turn off/on the TENS machine enabling stimulation when desired. For the human-to-human interaction application we additionally made use of an Enobio8 - an 8 channel EEG system to measure motor related neural activity of the sender subject to trigger involuntary forced movement in the receiver subject, using nerve stimulation.

Software Interface Design: We developed three software-based interfaces to validate our application scenarios:

1. **Navigation Interface:** A navigation app was developed in Android using the MapBox navigation API (Figure 1 (A), Figure 2) to provide real time data for turn by turn navigation (using announcement listeners of the JSON response generated from MapBox in real time) on a particular route selection from the current location. 3 meters prior to any significant left or right turns the application sent distinct byte commands over Bluetooth which was received by the Arduino to trigger the relay switch to enable/disable nerve stimulation and provide a nudge or forced movement in either the arms (L/R) or the legs (L/R) of the user at every turn according to the direction of the turn. We add-on to the work of Pfeiffer et al. [12] in that we develop a closed-loop mechanism using an actual navigation app similar to Google Maps and not a 'remote-controlled' button triggered stimulation while using a different technology of TENS and not EMS.



Figure 2: Navigation Nudges using our app and our hardware interface when user approaches a turn

2. **Feedback Gaming:** A circular track based car racing game was developed in Python (using PyGame) with bidirectional steer controls (Q key - left steer, P key - right steer). The default speed of the car could be varied between the range of 0-70 units (varying difficulty levels of control) and in our nudge trials the onset of turns and off-route steering sent data to the micro controller to stimulate radial nerves in the left or right hand leading to 'forced finger movements' to press the respective keys to steer the car in the correct direction - Figure 3 (A), (B).

3. **Human to Human Control:** Our human to human interface application was bifurcated into two parts: EEG acquisition of the sender subject to measure motor related movements and the corresponding electrical nerve stimulation in the receiver subject. For the EEG data acquisition, we made use of an 8 channel EEG system (Enobio8). We used a pre-existing off the shelf model provided by MNE Python library to identify motor related movements from EEG signals. The model comprised of the common spatial pattern (CSP) algorithm to initially extract features of data for unique brain states and alternatively simple amplitude based thresholding for motor movements in EEG. Effectively, a CSP algorithm bifurcates a multivariate neural signal into smaller components with highest variances and is a popular use technique in EEG motor imagery classification algorithms [18]. The model made use of secondary left- and right-hand movement EEG data (for our specified electrode locations) and the average spatial features for each of the groups (L/R) were computed to detect movements. A K-dimensional eigenvector comprising of the products of the data set and the average spatial filters for each of the two groups (L/R secondary data)

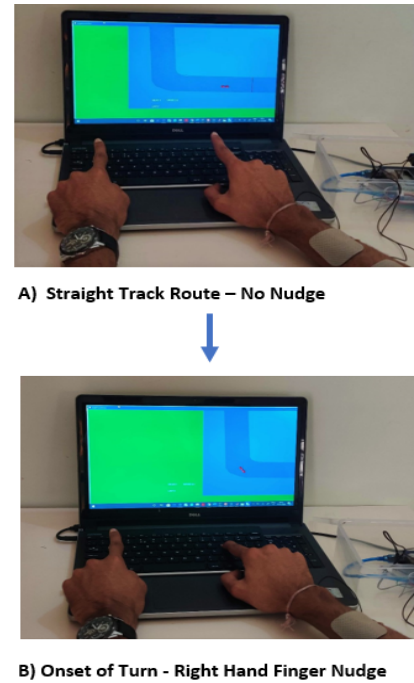


Figure 3: Gaming feedback nudges in our bi-control car steering game

was computed which then fed in our SVM model. Once the motor related neural activity was detected in the sender subject (using the real time SVM model) commands were sent to the micro-controller connected TENS unit to stimulate nerves in the receiver to trigger involuntary motor movements in the receiver subject.

Methods

Our preliminary study of 13 subjects (8 males, 5 females; ages 20-24) for navigation interface and feedback gaming study and 12 subjects (in pairs) for the human to human interface study aims to provide an early validation of our interface applications. The subjects were selected using opportunity sampling. An informed consent approval of each subject was taken and the participants were given debrief info and a feedback form at the conclusion of each study.

Navigation Study: A within subjects design was used whereby the independent variables were the position of the TENS electrode pads (radial nerve – arm, fibular nerve – leg) and three varying intensity of the stimulation (Intensity 4: 35mA, Intensity 5: 45mA, Intensity 6 – 55mA). All subjects were asked to walk along the same route six times for each combination condition (three levels of intensity x two different positions of electrode pads) after the hardware interface (placed in the user's bag) was setup and the Android Navigation app was turned on. The fixed route comprised 5 turns (a combination of both left and right turns). We validated the effectiveness of the nerve related motor movement at the onset of the turns. The dependent variables of the study were the nudge events triggered during the turns and a subjective evaluation of the

effectiveness of the nudges (different intensities, different limbic positions). Post study, subjects were asked to rate the three levels of intensity on a scale (1 – least preferred, 5 – most preferred) and their preferred location of stimulation for specific intensities.

Feedback Gaming Study: A short time interval based longitudinal within subjects design was used. Since we wanted to investigate improvement in performance due to 'forced event' nerve stimulation, we had a set of control trials and test-trials (preceded by trials with nerve stimulation) and therefore repeated observations per subject (longitudinal) were conducted in a single day to see if the number of mistakes were reduced post nerve stimulation trials. All the subjects were tested twice in a single day with a time gap of 6 hours between the control and test trials. The time gap was deliberate in order to reduce practice effects. In the morning, manual steering control trials were conducted on all the subjects whereby the subjects were asked to complete a lap of the car racing game (Q key – left hand index finger key press– left steering, P key – right hand index finger key press – right steering) twice in gaps of 5 minutes. The fixed variables were the default speed of the car set to 60 units (high speed) and the intensity of the stimulation (Intensity 5) and the dependent variables were the number of times the subjects lost control by steering away from the main track route.

A set of manual steering test trials were conducted in the evening which were preceded by 2 forced-event based trials per subject. In the forced movement trigger the subjects' left or right-hand finger were automatically stimulated to press the key when the subject was off-route or when a turn was about to occur. Two manual steering trials per subjects followed the nudge trials whereby the subjects were asked to steer the car using keys (without forced nudges). Figure 3 (A) shows how no forced nerve based motor events were stimulated when the car was on a straight route, however in Figure 3 (B) at the onset of a right turn, the game sent a nerve stimulation trigger to the right hand finger to press the 'P' key to turn the car. The number of times the subjects went off-track were again recorded. Using this interface design, we tried to validate whether the subject's performance in manual steering improved post forced-nudging. We made use of an intra-day longitudinal study and a smaller number of trials to reduce practice effects.

Human to Human interface Study: This study focused on validating the use of nerve stimulation techniques for human to human control applications. 12 subjects (in pairs) participated in the study where 6 subjects were used as senders while the other 6 were the corresponding receivers. The EEG system and cap were placed over the sender subjects to detect their brain activity associated with limbic movements while as the nerve stimulation system and TENS pads were connected to the receiver subjects to stimulate their ulnar nerve to trigger involuntary tri-finger movement (Figure 4 (A), (B)). Each sender subject was asked to move their arms five times in time intervals and corresponding forced-finger stimulation trigger events in the receiver were recorded, and the accuracy of the motor event detection algorithm was measured.

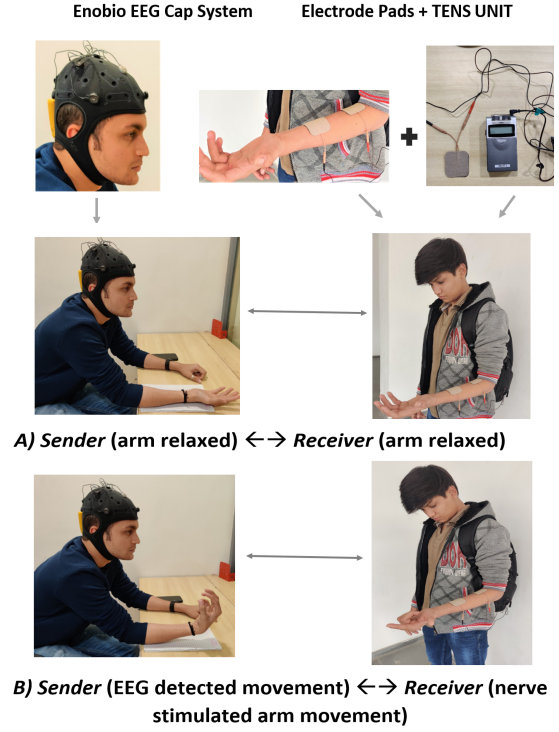


Figure 4: Human to Human interface using nerve stimulation and Enobio20 EEG

4 RESULTS

Navigation Interface: The two independent variables of the study were the position of the electrode pads (arms, legs) and the intensity of stimulation (Intensity 4, 5, 6). Figure 6 shows the line chart depicting the preferred stimulation intensity and preferred location for the subjects based on a scale rating. The X axis depicts levels of stimulation, the colour of the lines depicts the location while the Y axis depicts the average rating for each intensity x location combination averaged across all subjects.

Inferential one-way ANOVA analyses were carried out to understand if there was a significant difference between the means of the ratings for the three levels of stimulations for both (hand and leg). Since we compared more than two group of means, we used an ANOVA instead of individual t-tests. We assumed normality and continuity in mean rating data to simplify our analysis and therefore chose a one-way parametric ANOVA test.

Leg Stimulation Condition: Comparing means of stimulation level four, five, six for leg condition we found a significant difference in the means – $F(2,36) = 15.643$, $p < 0.01$. A Bonferroni post-hoc analysis revealed that mean rating for stimulation level four (2.5385 ± 1.126) and stimulation level six (2.0769 ± 0.76) was significantly lower than that of stimulation level five (4 ± 0.816). There was no significant difference between mean ratings for stimulation level four and six ($p = 0.62$) for the leg condition.

Arm Stimulation Condition: Comparing means of stimulation level four, five, six for arm condition we found a significant difference in the means – $F(2,36) = 13.565$, $p < 0.01$. A Bonferroni

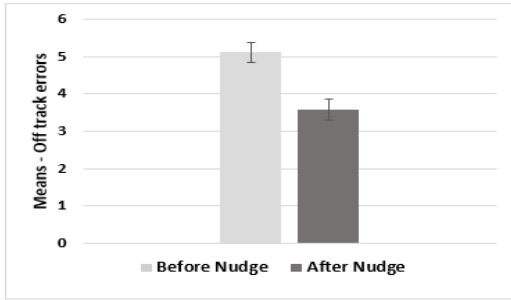


Figure 5: Bar plot depicting mean number of off-track errors per subject before and after nudge feedback trials (error bars = SEM)

post-hoc analysis revealed that mean rating for stimulation level six (1.538 ± 0.66) was significantly lower than rating of both stimulation level five (3.23 ± 0.73) and stimulation level four (2.461 ± 1.05). There was no significant difference between the ratings of stimulation four and five ($p = 0.07$) for the arm condition.

The graph and the inferential analysis depicts that the most preferred nudge stimulation intensity is level 5. The subjects stated that intensity level 6 nudge felt highly forced and abrupt and the effects felt even more exaggerated when the pads were placed in the arms as compared to the legs. Subjects reported that intensity level 4 although could be significantly felt, did not act a strong decision-alter nudge and therefore rating score for intensity level 5 could have been the highest. The subjects also reported that the same stimulation levels felt more exaggerated, jerkier when the electrode pads were placed in the arms as compared to the legs and therefore there was an average higher likeness score for electrodes placed on the legs.

Feedback Gaming: The fixed variables in this study were the default speed of the car set to 60 units (range 0 – 70) and the intensity of stimulation set to level 6. The number of times the subjects went off route in the before and after forced-event/nudge trial conditions were stored. We hypothesized an improvement in gaming performance and reduced number of off-track errors in post nudge feedback trials. A one tailed paired samples t-test was conducted to analyse whether there was a significant difference in the number of times each subject went off route before and after nudge trials. The bar plot in Figure 5 depicts the mean number of times each subject went off route in the before and after conditions. A significant difference was found between the means - $t(12) = -4.27$, $p = 0.00054$ - for pre-forced/nudge trigger and post-forced/nudge trigger event conditions with mean number of off route errors found to be significantly low in post-forced/nudge trigger conditions (mean = 3.58, SD = 1.01, SEM = 0.28) as compared to pre-forced/nudge conditions (mean = 5.115, SD = 0.98, SEM = 0.272).

Human to Human Control: Our human to human interface experiment comprised of five hand movement trials per subject across 6 subject combinations of senders and receivers. An evaluation of our CSP and SVM based classification algorithm to detect limbic motor activity using EEG reflected that the algorithm was able to

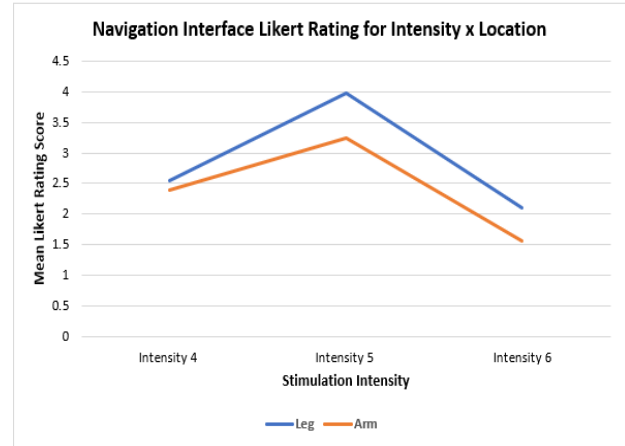


Figure 6: Line plot depicting mean rating score for each stimulation intensity and electrode pad location combination

detect 26 movement events (true positives) out of a total 30 actual movement events while the algorithm detected 4 false negatives and 3 false positives. Based on this we computed the algorithms accuracy = 78.7, precision = 89.6, sensitivity = 86.6. Since the purpose of this paper was an early validation of the use of nerve stimulation for human to human interaction and not model optimization for detection of motor related events using EEG, we haven't focused on any further validation or improvements in the machine learning model used for limbic EEG classification. In all situations where the model classified an event as a limbic motor activity in the sender, an involuntary forced movement was elicited in the little, ring and middle finger by nerve stimulation in the receiver.

5 CONCLUSION

In this paper we proposed the use of electrical nerve stimulation to generate nudges or forced action events for interesting closed-loop applications in HCI – nudging to aid navigation, feedback gaming improvement and human to human involuntary interaction. The interfaces are a work in progress but provide an early validation for our above-mentioned applications. Future work could involve the use of nudge-based navigation aid systems for the visually impaired, a concrete validation of nudge enhanced learning in motor related tasks such as rock climbing, video game controls, playing an instrument, nudge triggered involuntary car steering control in accident avoidance situations and the in use cases of neurofeedback.

A few limitations of our current work in progress study were the limited number of participants resulting in limited statistical analysis for our interfaces, the effect of practice in our feedback gaming interface – although we tried to limit significant practice effects by using a short interval based longitudinal study design. Further work could also focus on a more accurate and robust classification of motor movements (using EEG) to provide a more concrete use case for our human to human interaction and/or an extensive study of neurofeedback control using TENS. In our interface the distinction between a nudge and a 'forced event' is simply created

by the intensity of the stimulation, if it is a lower intensity stimulation it is a nudge simply because there is no automatic motor movement (hand or leg) elicited, however increasing the intensity would 'force' trigger motor movements and would no longer be a nudge. This would create some drawbacks associated with using nerve stimulation based 'closed-loop' nudges leading to a freedom of choice-effectiveness trade off. For example in building an immediate accident avoidance system, 'forced' nerve elicited movements would be much more effective since there would be no delay created by user reaction time, however an ethical dilemma arises since the driver no longer has a freedom of choice. Alternatively, if we used a nudge based accident avoidance system, the freedom of choice would lead to a delay in decision making and therefore reduce effectiveness in avoiding accidents. Therefore, further work could also discuss how to increase effectiveness of nerve stimulation 'nudges' without making them 'forced events'.

REFERENCES

- [1] Hazim Almuhiemi, Florian Schaub, Norman Sadeh, Idris Adjerd, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. 2015. Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 787–796.
- [2] Oswald Barral, Gabor Aranyi, Sid Kouider, Alan Lindsay, Hielke Prins, Imtiaz Ahmed, Giulio Jacucci, Paolo Negri, Luciano Gamberini, David Pizzi, et al. 2014. Covert persuasive technologies: bringing subliminal cues to human-computer interaction. In *International Conference on Persuasive Technology*. Springer, 1–12.
- [3] Wisnu Broto, Noor Suryaningsih, Fauzie Busalim, and Ane Prasetyowati. 2017. Impulse Of Triangle and Square Wave With TENS Methode Based On Microcontroller. In *Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence*. ACM, 285–288.
- [4] Ana Caraban, Evangelos Karapanos, Daniel Gonçalves, and Pedro Campos. 2019. 23 Ways to Nudge: A Review of Technology-Mediated Nudging in Human-Computer Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 503.
- [5] Ashley Colley, Aki Leinonen, Meri-Tuulia Forsman, and Jonna Häkkinen. 2018. EMS Painter: Co-creating Visual Art using Electrical Muscle Stimulation. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. 266–270.
- [6] Stephanie Coulthard, Ignace Hooge, Monique Smeets, and Elizabeth Zandstra. 2017. Nudging food into a healthy direction: The effects of front-of-pack implicit visual cues on food choice. *International Journal of Food Design* 2, 2 (2017), 225–240.
- [7] Ayaka Ebisu, Satoshi Hashizume, Kenta Suzuki, Akira Ishii, Mose Sakashita, and Yoichi Ochiai. 2017. Stimulated percussions: method to control human for learning music by using electrical muscle stimulation. In *Proceedings of the 8th Augmented Human International Conference*. 1–5.
- [8] Bart Engelen. 2019. Ethical criteria for health-promoting nudges: a case-by-case analysis. *The American journal of bioethics* 19, 5 (2019), 48–59.
- [9] Ernst Kruijff, Dieter Schmalstieg, and Steffi Beckhaus. 2006. Using neuromuscular electrical stimulation for pseudo-haptic feedback. In *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 316–319.
- [10] Melody Moore, Philip Kennedy, Elizabeth Mynatt, and Jennifer Mankoff. 2001. Nudge and shove: frequency thresholding for navigation in direct brain-computer interfaces. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*. ACM, 361–362.
- [11] Fabian Okeke, Michael Sobolev, Nicola Dell, and Deborah Estrin. 2018. Good vibrations: can a digital nudge reduce digital overload?. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 4.
- [12] Max Pfeiffer, Tim Dünne, Stefan Schneegass, Florian Alt, and Michael Rohs. 2015. Cruise control for pedestrians: Controlling walking direction using electrical muscle stimulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2505–2514.
- [13] Max Pfeiffer, Tobias Kröger, Jens Seifert, Sulaxan Somaskantharajan, Lukas Jahnich, Tobias Steinblum, Jan Speckamp, and Samuel Navas Medrano. 2019. WONDER-Enhancing VR Training with Electrical Muscle Stimulation. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.
- [14] Rajesh PN Rao, Andrea Stocco, Matthew Bryan, Devapratim Sarma, Tiffany M Youngquist, Joseph Wu, and Chantel S Prat. 2014. A direct brain-to-brain interface in humans. *PloS one* 9, 11 (2014), e111332.
- [15] John T Serences and Steven Yantis. 2006. Selective visual attention and perceptual coherence. *Trends in cognitive sciences* 10, 1 (2006), 38–45.
- [16] Markus Weinmann, Christoph Schneider, and Jan vom Brocke. 2016. Digital nudging. *Business & Information Systems Engineering* 58, 6 (2016), 433–436.
- [17] Daryl W Wiesman. 2006. The effects of performance feedback and social reinforcement on up-selling at fast-food restaurants. *Journal of organizational behavior management* 26, 4 (2006), 1–18.
- [18] Weibo Yi, Shuang Qiu, Hongzhi Qi, Lixin Zhang, Baikun Wan, and Dong Ming. 2013. EEG feature comparison and classification of simple and compound limb motor imagery. *Journal of neuroengineering and rehabilitation* 10, 1 (2013), 106.

Engineering Privacy-aware Smart Home Environments

Akshika Wijesundara
School of Computing and Communications
The Open University, UK
akshika.wijesundara@open.ac.uk

ABSTRACT

Smart homes (SmHs) integrate a number of user interaction technologies, which have the potential to cause privacy violations between occupants of the home either through inappropriate information disclosures or disturbances. One approach to protecting SmH occupants' privacy is to develop techniques for dynamically adapting the UI layer according to the context, as well as each occupant's preferences and capabilities. Therefore, the focus of this research is to develop a framework and set of algorithms which can adapt the configuration and behavior of UI layer in order to preserve the privacy of SmH occupants while maintaining usability.

CCS CONCEPTS

•Security and privacy~Human and societal aspects of security and privacy•Human-centered computing~Ubiquitous and mobile computing~Ubiquitous and mobile computing systems and tools

KEYWORDS

Adaptive user interfaces; End user privacy management; Smart home interfaces

ACM Reference format:

Akshika Wijesundara. Engineering Privacy-aware Smart Home Environments. In *Proceedings of ACM Symposium on Engineering Interactive Computer Systems, Sophia-Antipolis, France, 3 pages*. <https://doi.org/10.1145/3393672.3398643>

1 Introduction

The user interface (UI) layer of a Smart Home (SmH) enables occupants to interact with their home through a variety of devices, including smart speakers, TVs, smartphones and wearable devices, which are also shared by other occupants of the SmH. Because some of these UIs are shared (public UI) between multiple occupants of a SmH, their use can lead to violations of users' privacy. These users can also have conflicting privacy

requirements, UI preferences and different capabilities. The following hypothetical scenario is representative of a possible privacy violation scenario

Sally is an older lady who lives alone in her own SmH. She has a circle of support which comprise her adult son (Zack) and a care giver (Yasmin). Sally only shares her health data with her care giver and Sally's care giver prefers to use the smart phone as an alternative UI to the smart speaker.

One day while Yasmin is reviewing Sally's vital health measurements using the smart speaker, Zak enters the room. Sally's blood glucose levels have increased. The SmH senses the arrival of Zak and pauses the broadcast of Sally's data over the smart speaker and automatically switches to Yasmin's phone.

If the SmH UI layer was not able to adapt the broadcast of the health data, Sally's privacy preferences would have been violated, potentially causing embarrassment for her and Zack. Even though existing SmH UI frameworks support the functionality of the UI layer, none of them are able to preserve different privacy preferences of multiple SmH users while they are interacting with public UIs. Managing these conflicting and complex user requirements and adapting the UI layer to maximize satisfaction of these requirements poses a significant challenge. In order to address this, this research proposes an engineering framework to guide the construction and maintenance of the SmH UI layer. The proposed adaptive UI framework is capable of modelling different user's privacy preferences, user capabilities as well as UI preferences. The framework will provide algorithms for detecting potential privacy violations and generating suitable UI adaptations to maximize the satisfaction of users' privacy preferences as well as the usability of the system. This paper reports the research methodology planned to address the aforementioned research gap and the progress of the research conducted thus far.

2 Related Work

Privacy violations that happens at the SmH UI layer can be divided in to two major segments: (1) privacy violations that could happen outside the SmH due to information leakages to third party entities [1] [2] [3] [4] and (2) privacy violations that happen between SmH users [5] [6]. This research is focused on the latter type of privacy risk which is referred to as *Interpersonal Cyber Physical Privacy*.

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 third-party components of this work must be honored. For all other uses, contact the Owner/Author.
EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France
© 2020 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-7984-7/20/06.
<https://doi.org/10.1145/3393672.3398643>

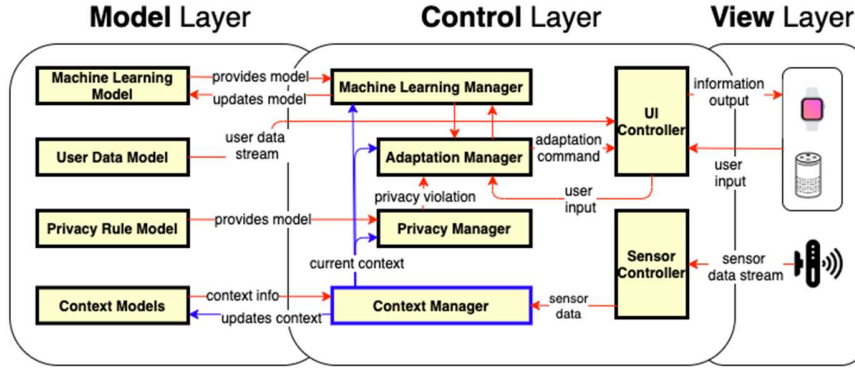


Figure 1: Privacy-aware Adaptive Smart Home Interface Framework

Frameworks such as Dynamo-AID [7], MASP [8] and GPII [9] have provided support for the development of multimodal adaptive user interfaces for SmHs. But none of them supports different privacy preferences of SmH users.

A framework for engineering privacy aware Smart Homes also requires modelling user preferences and capabilities. In order to represent privacy preferences, the Rei [10] policy language has been used and Virtual User Modelling and Simulation clusters are used to model users' capabilities and UI preferences [11].

3 Research question and methodology

The primary research question investigated is:

"How can smart home interfaces be engineered to adapt their configuration and behavior to preserve privacy and usability for users in multi-occupancy contexts?"

Models for the user preference and capabilities were developed through a literature review of modelling techniques, to synthesize a meta-model that captures the required knowledge. This meta-model is validated using a case study. The adaptation algorithms and associated the software architecture (Figure 1) will be evaluated in terms of scalability and performance using simulations. Finally, a lab-based user study will be conducted to explore the effectiveness of the framework in preserving privacy and usability in practical settings.

4 Evaluation

A within group study with 15 participants was conducted. The participants were given pictorial storyboards consisting of 2 hypothetical privacy violation scenarios. Each scenario had 3 variations of UI adaptations (automatic adaptive(AA), recommendation based adaptive(RBU) and non-adaptive(NA)) where participants were asked to rate the usability of each variation using the system usability score (SUS) instrument[13]. The order of the 6 scenarios were methodically changed while participants were randomly assigned to each variation of the study. This randomization minimized the possible learning effect. This was followed by a semi-structured interview to understand their reasoning for the usability rating provided. SUS scores were analyzed using one-way repeated measures ANOVA and followed

by a Tukey test. There was significant difference between AAUIs and NAUIs ($F(87, 2) = 5.002502, p = 0.00878$). Therefore, it can be claimed that automatic adaptive UI was better compared to non-adaptive UIs in delivering a usable system that persevered users' privacy.

Further a qualitative analysis was conducted to understand the reasoning behind the SUS scores. Participants preferred adaptive-UI (AAUI or RBAUI) over NAUI due to error prevention, flexibility and efficiency of use, aesthetic and minimalist design where some of the participants raised the concern over losing control and freedom of their interaction while using adaptive-UI. Even though, there was no significant difference between RBAUI and AAUI in protecting privacy as a whole, participants had divided opinions in picking their favorite variation. The discussion revolved around visibility of system status, user control and freedom, flexibility and efficiency of use, aesthetic and minimalist design and error prevention. Analysis of this data led to the conclusion that users might like to have more user control over less critical types of privacy such as disturbances, whereas they prefer stricter system enforced control in relation to information disclosure privacy violations.

5 Proposed framework

The model layer of the proposed framework (Figure 1), combines contextual information, user privacy rules, user data models with machine learning models that control the adaptation of the user interfaces in a SmH environment. These models were developed and evaluated using case studies derived from the STRETCH project [12], which is investigating how SmH technologies can support older adults to live independently in their homes. In order to support this adaptation the framework represents, user privacy preference in the Rei policy language[10] and together with everyone's preferences and capabilities using VUMS cluster[11].

The Rei language is used to represent privacy rules such as: Role "care receiver"'s health data can only be shared with people with the role "caregiver". In Rei, this becomes: `Has(P, right (accessing_health_data, X, [role(sally, care_giver, P), data(sally, health, X)], [assert(access(X))])`

Figure 1 shows the software architecture of the framework. SmH sensors updates the context manager of any context changes where context manager triggers the privacy manager to check for possible privacy violations. If there is a possible privacy violation, the privacy manager invokes the adaptation manager to adapt the UI to preserve the privacy of the SmH users while maintaining the usability of the system. Feedback on this adaptation from the user will also be captured and used to update the machine learning model. Later on, the machine learning model will learn the user's privacy and UI preferences to enable fully autonomous adaptation. The initial evaluation of the adaptations generated from the framework was conducted through a usability study that adopted a 'Wizard of Oz' approach based pictorial storyboards as described in §4. The segments related to context model, context manager, sensor control and machine learning models were simulated, and the results of the hypothetical scenarios were depicted through pictorial story boards to be evaluated by users.

6 Conclusion and Future Work

The usability study demonstrated that adaptive UIs were capable of preserving privacy of SmH occupants with better usability. Future work will conduct simulation studies to explore the performance of the adaptive architecture, as well as a lab-based user study to evaluate the framework realistic settings. The study would explore privacy violating scenarios that could be avoided using adaptive SmH UIs having a strong emphasis on qualitative analysis to understand user perceptions.

ACKNOWLEDGMENTS

This work is funded by The Open University, and supervised by Professors Arosha Bandara, Blaine Price and Bashar Nuseibeh.

REFERENCES

- [1] C. Jackson and A. Orebaugh, 'A study of security and privacy issues associated with the Amazon Echo', *International Journal of Internet of Things and Cyber-Assurance*, vol. 1, no. 1, pp. 91–100, Jan. 2018, doi: 10.1504/IJITCA.2018.090172.
- [2] S. K. Das, K. Kant, and N. Zhang, *Handbook on Securing Cyber-Physical Critical Infrastructure*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.
- [3] R. Alharbi and D. Aspinall, 'An IoT analysis framework: An investigation of IoT smart cameras' vulnerabilities', in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, 2018, pp. 1–10, doi: 10.1049/cp.2018.0047.
- [4] X. Zheng, Z. Cai, and Y. Li, 'Data Linkage in Smart Internet of Things Systems: A Consideration from a Privacy Perspective', *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, Sep. 2018, doi: 10.1109/MCOM.2018.1701245.
- [5] C. Kray, G. Kortuem, and R. Wasinger, 'Concepts and issues in interfaces for multiple users and multiple devices', in *Workshop on Multi-User and Ubiquitous User Interfaces (MU3I) at UII 2004*, Funchal, Madeira, Portugal, 2004.
- [6] M. J. Kraemer, 'Preserving Privacy in Smart Homes: A Socio-Cultural Approach', in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, Montreal QC, Canada, 2018, pp. 1–4, doi: 10.1145/3170427.3173018.
- [7] T. Clerckx, K. Luyten, and K. Coninx, 'DynaMo-AID: A Design Process and a Runtime Architecture for Dynamic Model-Based User Interface Development', in *Engineering Human Computer Interaction and Interactive Systems*, Berlin, Heidelberg, 2005, pp. 77–95, doi: 10.1007/11431879_5.
- [8] M. Blumendorf, S. Feuerstack, and S. Albayrak, 'Multimodal user interfaces for smart environments: the multi-access service platform', in *Proceedings of the working conference on Advanced visual interfaces*, Napoli, Italy, 2008, pp. 478–479, doi: 10.1145/1385569.1385665.

- [9] C. Loitsch, G. Weber, N. Kaklanis, K. Votis, and D. Tzovaras, 'A knowledge-based approach to user interface adaptation from preferences and for special needs', *User Model User-Adap Inter*, vol. 27, no. 3–5, pp. 445–491, Dec. 2017, doi: 10.1007/s11257-017-9196-z.
- [10] L. Kagal, T. Finin, and Anupam Joshi, 'A policy language for a pervasive computing environment', in *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, 2003, pp. 63–74, doi: 10.1109/POLICY.2003.1206958.
- [11] N. Kaklanis *et al.*, 'Towards standardisation of user models for simulation and adaptation purposes', *Univers. Access Inf. Soc.*, vol. 15, no. 1, pp. 21–48, Mar. 2016, doi: 10.1007/s10209-014-0371-2.
- [12] 'STRETCH - Socio-Technical Resilience for Enhancing Targeted Community Healthcare'. [Online]. Available: <https://stretchproject.org/home>. [Accessed: 25-Nov-2019].
- [13] P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, *Usability Evaluation in Industry*. CRC Press, 1996.

Towards Supporting Reproducibility of Experimental Studies in GUI Visual Design

Nicolas Burny

nicolas.burny@uclouvain.be

Université catholique de Louvain (UCLouvain), Louvain Research Institute in Management and Organizations (LouRIM)
Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium

ABSTRACT

Graphical User Interfaces are the most common way of interaction with the devices we use in our everyday life. Given the important and long-lasting impact that the visual design of GUIs has on User Experience, its experimental study is of high importance. However, this activity suffers from a lack of reproducibility of experimental results due to the significant amount of time and resources to conduct such experiments and create datasets. To address this problem, this thesis aims at developing an application which purpose is to facilitate the construction of datasets in the context of experimental studies of GUIs. The application parameterizes the design of experimental studies related to GUIs and automates various steps in order to facilitate their deployment and to foster their reproducibility. We explain the research approach, the workflows and features underlying the application. Finally, we discuss the current state of the thesis and the future work to be achieved.

CCS CONCEPTS

• **Human-centered computing** → Graphical user interfaces; User studies; *User interface design*; • **Computer systems organization** → Cloud computing.

KEYWORDS

Reproducibility; Experimental Studies; Graphical User Interfaces; Measures, Visual Design

ACM Reference Format:

Nicolas Burny. 2020. Towards Supporting Reproducibility of Experimental Studies in GUI Visual Design. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3393672.3398644>

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7984-7/20/06.

<https://doi.org/10.1145/3393672.3398644>

1 INTRODUCTION

Graphical User Interfaces are the most common way of interaction with the devices we use in our everyday life and their main purpose is to convey information to the user. However, while the content is of high importance, the way the information is brought to the user is equally important. Various facets of GUIs such as visual aesthetics or usability have been studied extensively in the last decades. Visual appreciation, for example, is one of the first interactions people have with an interface when they first interact with it. Some studies report that this effect occurs within the first 500 milliseconds of the interaction [4]. The impact of visual aesthetics on other dimensions such as credibility, usefulness or performance has also been demonstrated [9]. GUIs have thus an important and long-lasting impact on User Experience (UX) and their design is key for developers and designers.

One particular research direction that grew considerably in the last few years is the experimental study of GUIs visual design characteristics through various types of metrics or measures¹, such as for aesthetics [2, 13], for visual techniques [11]. Despite its rapid development and the growing interest in the field, the study of GUIs visual features encounters some problems that hinders its development.

Motivations

Experiments related to GUIs visual design make heavy use of data to study the correlation between features of user interfaces and dimensions of UX [5] or to develop models which goal is to predict the user score for a particular facet of UX [8]. This research field suffers from data-related problems due to the very small number of datasets publicly available and the difficulty to create new data sets. The problem of availability of data sets, or the lack thereof, coupled to the tedious process to build them hinders the reproducibility, repeatability and verifiability of experiments. Not only it is vital to support various experimental methods for studying the different facets of GUIs visual design, but also these methods should be replicable [7].

¹Although the Software Testing community today recommends using the term “measure” instead of “metric”, the two terms will be used alternatively as they are considered equivalent in our context.

This is a major problem because science is a cumulative process, in which facts are confronted with other facts and new experimental results consolidate theories or, on the contrary, invalidate them. Reproducibility is the cornerstone of the cumulative science and is a requirement in many research settings in order to assess the value of scientific claims. Many attempts have been made to raise awareness of the importance of reproducibility, to formalize and structure reproducibility in science [7], including in Computer Science and Human-Computer Interaction (HCI).

2 RELATED WORK

Reproducibility is a matter of habits and good practices on the part of the researcher, but it is also related to the tools and technologies that makes the processes efficient and realistic. Considering the experimental study of GUIs, there exists several tools allowing for the partial automation of the dataset construction process [1, 6, 8, 12]. These tools mainly focus on the evaluation of GUIs by either computing measures (measure-based evaluation) or gathering data about users perception of several GUIs features (user-based evaluation) and a lot of tedious tasks such as building experiments or capturing screenshots are left to the experimenter. The architecture of these tools also create issues regarding their scalability and extensibility.

The tools existing in the field of GUIs evaluation are based on a Service-Oriented Architecture (SOA), where a service is defined as the encapsulation of a series of components in order to provide a discrete business functionality. Traditional SOA-based applications are built on a n-tier architecture where each of the presentation, processing and data management layers are logically and physically separated. But recently, the emergence of more flexible business models have created a demand for architectures that can scale more freely than the multi-tier model [3].

3 PROBLEM STATEMENT

In the light of the raised issues in the field of experimental studies on GUIs visual design, we aim at answering the following question:

Is it possible to improve the reproducibility of experimental studies in the field of GUIs visual design by facilitating the construction of datasets with the help of a set of tools allowing for the automation of the various parts of the dataset building process ?

In this thesis, we propose a tool aiming at facilitating the construction of datasets in the field of GUIs experimental studies by automating the various parts of the dataset construction process, including the capture of GUIs screenshots, the design and deployment of experiments aiming at collecting data regarding users evaluation of various features of

GUIs and the computation of measures on captured interfaces. As illustrated in figure 1, the tool supports an experimental workflow (figure 1) and setups an environment in which researchers can easily perform experiments regarding the study of GUIs visual design.

4 RESEARCH APPROACH

We structure the main workflow of scientific research through an experimental method into three cycles (see figure 1): data collection cycle, experiment cycle, and development cycle, which are further refined in the next subsections.

Experiment Cycle

According to this procedure, researchers first have to state a research question and hypothesis regarding the topic of interest (figure 1, 1b & 2b). In the context of experimental study of GUIs, the application we are building in this thesis can be used to automate the process starting from the data collection step (figure 1, 3b). Our tool allows researchers to build experiments based on a template where they can specify introductory sections, pre- and post-test surveys used to gather user socio-demographic data and the variables they want users to evaluate on a set of GUI screenshots.

Along with the gathering of user-related data, the application allows experimenters to compute measures on GUIs. Measures can be obtained from any source, provided that the source code of the measure and its required dependencies are available. The most common sources of measures are Git repositories hosting the source code of the measures and their dependencies. The measures can be written in any language supported by the tool.

Once enough user-related data have been collected through the data collection cycle described hereunder, the experimenter can proceed to the data analysis part of the experimental workflow and report the results of the analysis (figure 1, 4b to 6b). These steps must still be done externally by the mean of modeling tools or languages such as Python, R, SPSS, etc. as the purpose of the application is not to provide a data analysis framework to researchers.

Experimenters can easily replicate their experiments them in various contexts (e.g., in different languages). The ease of replication of experiments allows experimenters to collect more data and to perform experiment at larger scale than if they had to build experiments manually each time they want to collect data in a different context. This is also particularly the case for very large models in model-based user interface design [10].

Data collection cycle

The participation of an user to an experiment follows the workflow depicted in the left-side diagram of figure 1. First of all, participants are displayed the introductory sections as

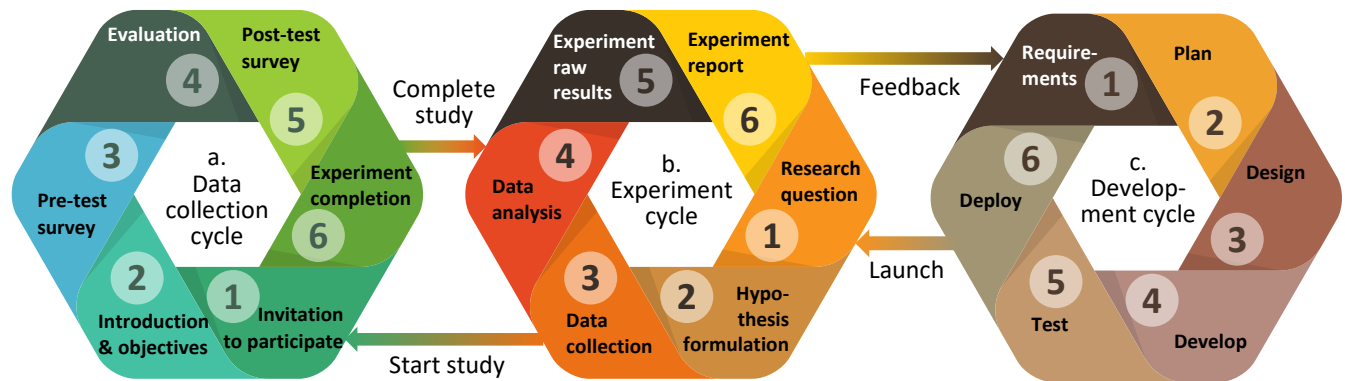


Figure 1: Link between experimental and development workflows.

specified by experimenters (figure 1, 1a & 2a). They have then to answer the questions of the pre-test survey (if applicable) (figure 1, 3a) before taking part in the actual experiment. During the experiment (figure 1, 4a), participants are asked to evaluate visual design of interfaces according to specific dimensions specified by experimenters (beauty, complexity, balance, etc). After all the GUI screenshots have been evaluated, participants are invited to answer a few questions being part of the post-test survey (figure 1, 5a) before finishing the experiment (figure 1, 6a).

Development cycle

At some point, the existing functionalities of the application may not be sufficient anymore to ensure that the working environment provided by the application meets the needs of the researchers. In such situation, further developments may be required. The architecture of the application allows for the development of new modules easily given the low coupling of the various components. The new modules can be developed according to a development cycle based on any methodology (figure 1, 1c to 6c). Experimenters can iterate as much or as little as needed in the development cycle before coming back to the experimental workflow with the new features implemented by the newly developed modules. Moreover, each module can be developed using its own technological stack, which increases the flexibility and ease of developments.

Datasets as Union of Data Sources

The main goal of the application described here is to facilitate the construction of datasets in order to increase the reproducibility of experimental studies in the field of GUIs visual design. Datasets must be seen as the union of various data sources with the GUIs being the pivot entity, e.g. data related to the evaluation of GUIs screenshots by users

or related to the computation of measures on GUIs screenshots. Researchers may be interested only in a subset of the data sources provided by the application. For example, researchers may be interested only in collecting user-related data while others may be interested only in measures-related data. In the application, each data source can be downloaded separately, which allows users to build the dataset tailored to their needs by joining the various data sources. As the application evolves and with the development of additional modules, more data sources may be available in the future.

Reproducibility of Experiments

In order to be able to reproduce an experiment, researchers must be able to deploy the same experimental setup as the original team or person that designed the experiment. However, such reproduction of an experimental setup may be a tedious and error-prone endeavour depending on the resources and information available about the original experiment. Our application allows users to download the complete experimental setup as a single configuration file that can easily be shared with other researchers. The application also allows users to rebuild the complete experimental setup contained in the configuration file by uploading it into the application. With such mechanism, we allow researchers to replicate a specific experimental setup with minimal effort. Experimental setup files are not to be confused with datasets as they do not contain any collected data from data sources but they contain data allowing users to deploy experiments in order to collect data according to the same experimental setup as the original experiment. The development of new modules may allow users to deploy various kinds of experiments. These modules should implement this type of mechanism in order to ensure that the related experimental setups may be shared and reproduced by different teams with minimal effort.

5 CURRENT STATE OF DEVELOPMENT AND FUTURE WORK

The application is currently under active development, a first version being deployed and under testing. Regarding the languages supported for measures computation, Python is currently the only language supported by the application but Java and Javascript will be available for measure computation in future versions of the application. The current modules under development allow for the capture of screenshots of websites GUIs and their evaluation by users through experiments. More modules will be developed in the future by our own team and hopefully by the community once the application will be officially released in open-source.

6 CONCLUSION

The field of Human-Computer Interaction suffers from a reproducibility crisis and the experimental studies of visual design of Graphical User Interfaces makes no exception. Current tools in this field have flaws that make them unsuitable for supporting reproducible experimental setups. The first goal of this thesis is the conception of an application allowing for the automation of the experimental design process, from the capture of screenshots of GUIs to the computation of measures on these interfaces. The application have been design with modularity in mind in order to facilitate the development of new features. Several modules of the application are currently under development and the application will be released in open-source in order for the community to be able to participate at the conception of new modules.

REFERENCES

- [1] Maxim Bakaev, Sebastian Heil, Vladimir Khvorostov, and Martin Gaedke. 2019. Auto-Extraction and Integration of Metrics for Web User Interfaces. *Journal of Web Engineering (JWE)* 17 (03 2019), 561–590. <https://doi.org/10.13052/jwe1540-9589.17676>
- [2] Sophie Dupuy-Chessa, Yann Laurillau, and Eric Céret. 2016. Considering Aesthetics and Usability Temporalities in a Model Based Development Process. In *Actes de La 28ième Conférence Francophone Sur l'Interaction Homme-Machine* (Fribourg, Switzerland) (IHM '16). Association for Computing Machinery, New York, NY, USA, 25–35. <https://doi.org/10.1145/3004107.3004122>
- [3] Martin Fowler James Lewis. 2014. *Microservices, a definition of this new architectural term*. <https://www.martinfowler.com/articles/microservices.html>
- [4] Gitte Lindgaard, Gary Fernandes, Cathy Dudek, and Judith M. Brown. 2006. Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour and Information Technology*, 25(2), 115–126. *Behaviour IT* 25 (03 2006), 115–126. <https://doi.org/10.1080/01449290500330448>
- [5] David Chek Ling Ngo, Lian Seng Teo, and John G. Byrne. 2003. Modelling Interface Aesthetics. *Inf. Sci.* 152, 1 (June 2003), 25–46. [https://doi.org/10.1016/S0020-0255\(02\)00404-8](https://doi.org/10.1016/S0020-0255(02)00404-8)
- [6] Antti Oulasvirta, Samuli De Pascale, Janin Koch, Thomas Langerak, Jussi Jokinen, Kashyap Todi, Markku Laine, Manoj Krishthombuge, Yuxi Zhu, Aliaksei Miniukovich, Gregorio Palmas, and Tino Weinkauff. 2018. Aalto Interface Metrics (AIM): A Service and Codebase for Computational GUI Evaluation. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings* (Berlin, Germany) (UIST '18 Adjunct). ACM, New York, NY, USA, 16–19. <https://doi.org/10.1145/3266037.3266087>
- [7] Roger D. Peng. 2011. Reproducible Research in Computational Science. *Science* 334, 6060 (2011), 1226–1227. <https://doi.org/10.1126/science.1213847>
- [8] Katharina Reinecke and Krzysztof Z. Gajos. 2014. Quantifying Visual Preferences Around the World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (CHI '14). ACM, New York, NY, USA, 11–20. <https://doi.org/10.1145/2556288.2557052>
- [9] N Tractinsky, A.S Katz, and D Ikar. 2000. What is beautiful is usable. *Interacting with Computers* 13, 2 (2000), 127 – 145. [https://doi.org/10.1016/S0953-5438\(00\)00031-X](https://doi.org/10.1016/S0953-5438(00)00031-X)
- [10] Jean Vanderdonckt and Pierre Berquin. 1999. Towards a Very Large Model-Based Approach for User Interface Development. In *1999 User Interfaces to Data Intensive Systems, UIDIS 1999, Edinburgh, Scotland, September 5-6, 1999*, Norman W. Paton and Tony Griffiths (Eds.). IEEE Computer Society, 76–85. <https://doi.org/10.1109/UIDIS.1999.791464>
- [11] Jean Vanderdonckt and Xavier Gillo. 1994. Visual Techniques for Traditional and Multimedia Layouts. In *Proceedings of the ACM International Workshop on Advanced Visual Interfaces* (Bari, Italy) (AVI '94). Association for Computing Machinery, New York, NY, USA, 95–104. <https://doi.org/10.1145/192309.192334>
- [12] Mathieu Zen and Jean Vanderdonckt. 2014. Towards an evaluation of graphical user interfaces aesthetics based on metrics. In *Proceedings of the IEEE Eighth International Conference on Research Challenges in Information Science* (Marrakech, Morocco, May 28-30, 2014) (RCIS '14). 1–12. <https://doi.org/10.1109/RCIS.2014.6861050>
- [13] Mathieu Zen and Jean Vanderdonckt. 2016. Assessing User Interface Aesthetics Based on the Inter-Subjectivity of Judgment. In *HCI 2016 - Fusion! Proceedings of the 30th International BCS Human Computer Interaction Conference, BCS HCI 2016, Bournemouth University, Poole, UK, 11-15 July 2016 (Workshops in Computing)*, Shamal Faily, Nan Jiang, Huseyin Dogan, and Jacqui Taylor (Eds.). BCS. <http://ewic.bcs.org/content/ConWebDoc/56903>

Method for Developing Virtual Reality Applications for Cognitive Intensive Training Tasks

Ketoma Vix Kemanji

Heilbronn University of Applied Sciences

Heilbronn, Germany

Université catholique de Louvain

B-1348 Louvain-la-Neuve, Belgium

ketoma-vix.kemanji@hs-heilbronn.de, ketoma-vix.kemanji@uclouvain.be



Figure 1: Virtual Reality for Skill and Behaviour Training

ABSTRACT

Virtual Reality applications have become popular over the last decade, gaining significant interest in industry and research. While the technology has its roots in the entertainment industry, there is an increase of research experiments and pilot studies where the technology is applied in professional context for behaviour and skill training. However, the implementation of effective training systems is a complex task; the development process remains largely ad hoc because there are no sufficient tools that appropriately support the development process. This thesis proposes a method for developing Head Mounted Display based virtual reality training applications for cognitive intensive training tasks. The method extends the concept of developing a parametrized system with run-time modifiable attributes and provides the necessary toolkit for the application implementation. The primary target user group are the developers and by extension the designer and training administrator.

CCS CONCEPTS

• **Software and its engineering** → **Software design engineering**; *Virtual worlds training simulations*; • **Human-centered computing** → *Virtual reality*.

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7984-7/20/06.

<https://doi.org/10.1145/3393672.3398645>

KEYWORDS

Virtual Reality; Training; Development Method; VR Framework

ACM Reference Format:

Ketoma Vix Kemanji. 2020. Method for Developing Virtual Reality Applications for Cognitive Intensive Training Tasks. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3393672.3398645>

1 INTRODUCTION

There is increasing development and research interest in the application of virtual reality (VR) technology in professional context for skill and behavior training [5]. With this increased interest has arisen the need for domain experts and training administrators, who often lack technical skills, to be able to personalize and modify attributes of the training system during use [9]. However, while some effort has been made to support the process, much of the development process of such training applications is still typically ad hoc [9] and there is no specific systematic methodology to support the process of developing such parametrized VR training systems.

Tools for developing VR training applications mainly stem from the traditional game development industry. While these tools have recently been adapted to developing VR content, the development of professional training-oriented VR applications is still considered a complex endeavor [4]. Third party frameworks have been developed which are integrated into the traditional game development tools [14], and other domain specific tools have been developed. However, there still does not exist tools to support the development

of parametrized VR training systems with the traditional and more popular game development tools.

A requirement of VR training applications is monitoring of trainee performance [10], evaluation of the cognitive requirements of the system and its effect on learning performance [1]. Training administrators require information and statistics on trainee performance during the training exercise to be able to personalize and optimize the training scenario [7]. While there exists some commercial tools, there is no open frameworks available for implementing monitoring and recording of trainee performance across different measures and properties of the training scenario in the more commonly used traditional game engines.

2 RELATED WORK

Previously proposed methods for designing parametrized, run-time modifiable VR training applications focus on separating expert and domain knowledge from the virtual environment and interactive objects. These methods allow specification of the virtual environment by the expert using a domain specific language and supporting tool to specify the semantics and ontology. Solutions in this category include VR-Demo [13], where a designer/domain expert defines the ontology of domain knowledge which is mapped to objects in the virtual environment and also carries semantic data about the virtual environment. In the EAST project [9], and the collaborative virtual environment by Barange [2], the authors utilize a domain specific language specification of the virtual environment and the MASCARET model [3, 4]. As part of the EAST project, Saunier *et al.* propose a methodology [9] for designing virtual environments for training. This ensures that the final environment appropriately reflects the professional real-world environment and the implementation choices of the computer scientist do not significantly impact the environment.

These methods encode the expert and pedagogic knowledge in a formal definition, which are used to instantiate the training scenario. Building on the concept of separation of roles, and inspired by previous methods: SENDA [8] and DeSMiR [12], this thesis promotes:

- Encoding expert knowledge in the requirements and design process.
- Focusing more on VR specific parameters of the training system.
- Simplifying the usability of training scenario personalization by means of simplified user interface which exposes modifiable attributes.
- Integrating performance monitoring of system attributes.

3 RESEARCH STATEMENT

This thesis addresses the issue of defining a method for developing head-mounted-display based VR applications for cognitive intensive training tasks, composed of training model with parameters, a step-wise approach for supporting this development, and a framework enacting the application of this approach. This method is applied on two case studies: a single user assembly of physical parts and virtual teaching/public speaking training.

4 METHODOLOGY

Based on the literature search model described by Zekowitz *et al.* [15] and research through design model [11], this thesis is implemented using an iterative methodology. The proposed method is developed based on state-of-the-art research literature and case studies. Fig. 2 shows an overview of the methodology. First, a review of the literature is conducted, this leads to the identification of relevant themes for implementing VR training applications. Five (three general, and two planned application specific) experimental Case Studies (CS) are conducted based on the identified themes, in the following application areas.

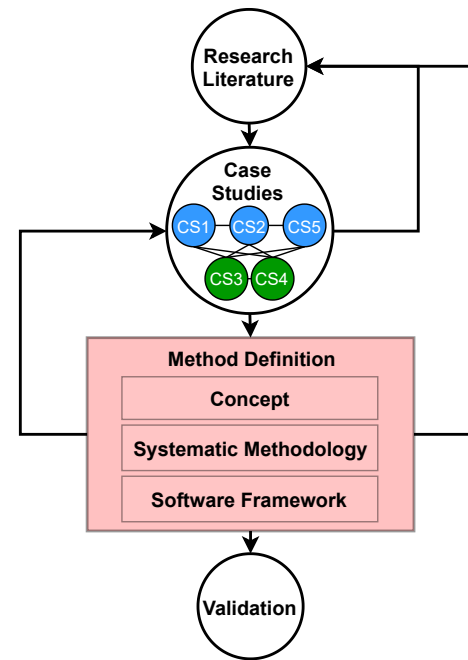


Figure 2: Methodology Overview.

General Case Studies (Completed)

- (1) CS1: Multi-kinect walking in place and VR application in retail.
- (2) CS2: Effect of Task Difficulty (Cognitive Load) and Presence on Virtual Object Perception.
- (3) CS5: User-Centered Gesture Interaction Design for VR: The Case of a Virtual Showroom.

Application Specific Case Studies (In progress)

- (1) CS3: Virtual Interfaces and Single User Assembly Training Tasks in Virtual Reality.
- (2) CS4: Virtual Reality Application in Teaching and Presentation Training.

Components of the framework are implemented in each case study, and the results of the experimental case studies each have implications on the development of the method.

5 INTERMEDIATE RESULTS

The intermediate results of this thesis includes a set of case-study experiments and results – Fig. 2, and the contribution of these experiments to the definition of a “Systematic Methodology”, and implementation of a “Software Framework Tool” – Fig.4 for VR training application development.

5.1 Case Studies and Results

The case studies introduced in Section 4 are conducted as part of this research. These case studies inform the design and implementation of the method and software framework. The general case studies: CS1, CS2 and CS5, provide general guidance on the implementation of VR experiences.

CS1 investigates the use of walking-in-place as a navigation technique for large virtual environments. The results of this case study indicate that suitability of navigation and interaction techniques are user and scenario specific [6]. This necessitates the integration of alternative virtual interfaces and interaction techniques in the design on the framework.

CS2 investigates the effect of cognitive load (task difficulty) and presence on the use of virtual objects for navigation guidance in virtual environments. The results of this case study show that impose different levels of cognitive load and have an effect on task performance. This informs the adaptable design of training and implementation of metrics for evaluating performance under different configurations of the VR training system.

CS5 investigates the design of user-defined natural gestures for interaction in virtual environments. This case study develops a taxonomy of user defined gestures for multi-modal interaction in virtual environments. These results will guide a future integration of natural gesture interaction modalities in the method and framework.

The application specific case studies CS3 and CS4 are planned to particularly inform the design of VR training applications for assembly of physical parts and teaching/public speaking training. The application specific case studies are designed and implemented using a modular adaptable technique using components of the VRTrain framework as discussed in the following sections.

5.2 Systematic Method

The systematic approach defines steps for developing virtual reality applications for training. The systematic method defines, tasks, roles, and workproducts using the UML notation as specified in the Software & Systems Process Engineering Meta-Model 2.0 (SPEM 2.0) specification of the Object Management Group. Fig. 3 shows an overview of the main Roles, Tasks and Work Product definitions. This Method Content definitions are independent of any specific process can be instantiated and used in any development process. The main Tasks, Roles and Work Products are described below:

- (1) Define Object Behaviours: This task is fulfilled by a domain expert with professional knowledge of operation of the specific application domain and the specific behavioural characteristics that virtual objects should exhibit. This task is performed together with the VR engineer who understands

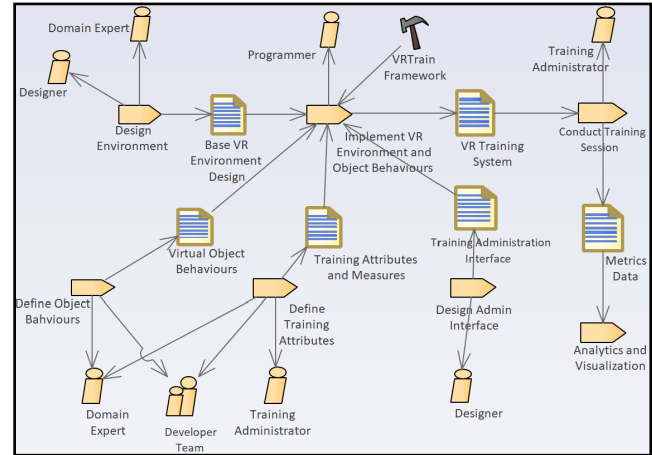


Figure 3: Systematic implementation method for VR training applications.

how and what behaviours can be simulated in a virtual environment. This Work Product of this task is a specification of virtual objects and their behaviours.

- (2) Define Training Attributes: Training attributes are include aspects of the training and simulation that can be varied by the training administrator. This task is fulfilled by the Training administrator, a Domain Expert and the VR engineer. The Work Product of this task is a specification of the Training Attributes and Measures (Metrics) used for tracking each of the specified attributes.
- (3) Design Virtual Environment & Training Administrative interface: The Work Product of these tasks are the Base Design of the VR Environment and the Training Administration Interface Design respectively. The Designer is primarily responsible for this task.
- (4) Implement VR environment and Object Behaviours: The programmer primarily fulfils this task using the Work Product output from the other tasks. Execution of this task is the primary focus of this method and framework. The programmer implements the virtual environment, virtual object behaviours, attributes and metric trackers. The VRTrain framework is used to fulfill this task as discussed in Section 5.3. The Work Product Out of this task is the VR training system.
- (5) Conduct Training session: The Training Administrator is sets-up a training session through an administrative interface. Setting up a training session consists of defining the virtual object behaviours, performance measures, metrics and configuring adaptable parameters for the specific training needs.

5.3 Software Framework Enacting Method

To enact the method described in Section 5.2 the VRTrain framework is developed to support the implementation step of the development process. The VRTrain framework is designed to be extendable to support a wide range of virtual reality training applications. However, in this thesis the focus is on development of VR training

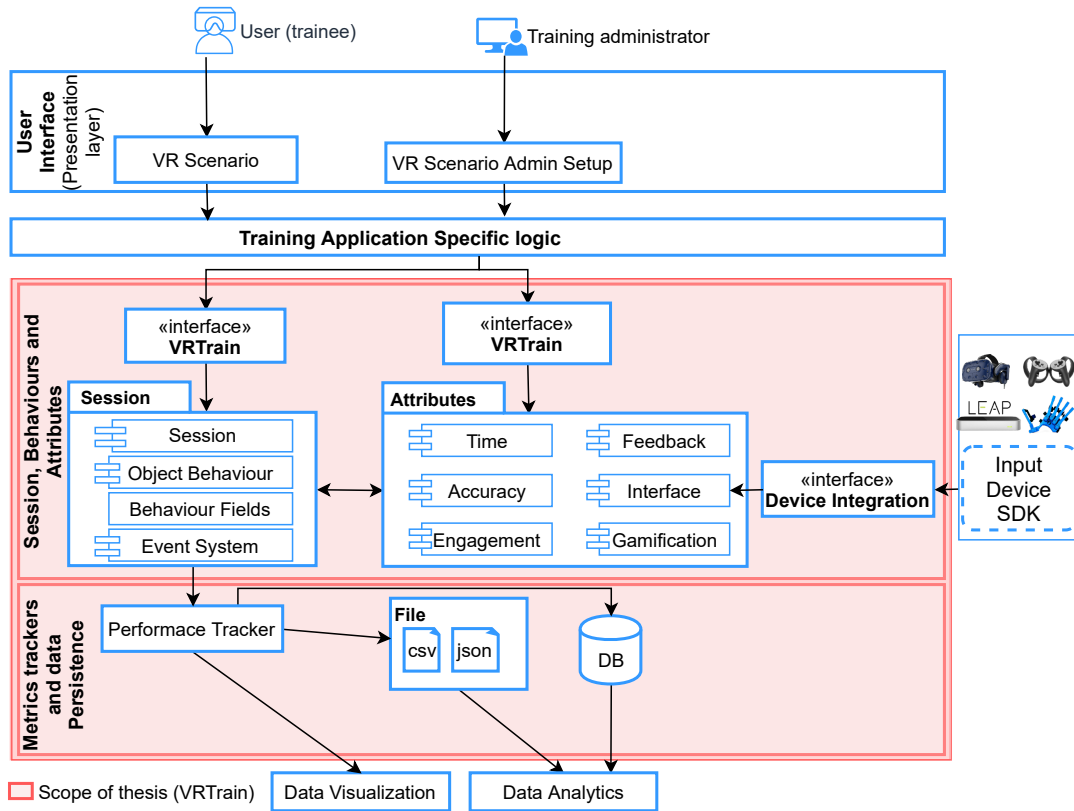


Figure 4: Software Framework Architecture.

applications for Assembly of physical parts and teaching & public speaking training in VR. For this two application scenarios, this thesis focuses on implementing object behaviours and modules that support, attributes, measures, and trackers for training feedback, virtual interface and gamification.

Fig. 4 shows an overview of the VTRain framework architecture. The framework runs on the scripting layer of the Unity game engine. The scope of the framework consists of two main layers: the Session, Behaviours and Attributes layer, and the Metrics trackers and data persistence layer. The top layer consists of the VR Scenario for training and the Administrative interface for the training administrator. The application specific logic interfaces between this presentation layer and the framework layers.

- (1) Session, Behaviours and Attributes: This layer consists of two packages for managing training sessions, and configuring session attributes and measures. The session packages contains modules, classes and interfaces for defining objects, object behaviours, and exposing adaptable/modifiable behaviour properties through behaviour fields. Behaviour fields are then exposed to the training administrator through the administration interface. This package also contains a module to help developers easily integrate an event manager and configurable events that are fired during a training session. The Attributes package contains modules, classes and interfaces that enable integration of configurable training

session attributes (feedback, virtual interfaces) and performance measures (time, accuracy, engagement).

- (2) Metric trackers and data persistence: This layer consists of modules, and classes for recording configured performance measures and metrics and persisting this data in flat files or a database. The recorded data can be exported for analysis within an external analytics tool (e.g. exporting data in csv format and importing into IBM SPSS for analytics) or directly streamed into an external visualization tool.

6 CONTRIBUTIONS

This thesis contributes a method for engineering VR training applications for cognitive intensive tasks as well as the needed development toolkit to implement the software according to the proposed method. This method contribution has three main components. First, a concept, which emphasises the interdisciplinary nature of implementing effective VR training applications. Second, a systematic methodology: a set of "Method Definitions" specified using SPEM 2.0 stereotypes – that defines a systematic approach for implementing VR training applications with implications for the requirements, design and use phases. And, third, a software framework tool, which supports the systematic methodology, targeted at developers at the implementation phase of the development process. Fig. 4 shows an overview of the VTRain framework software architecture, the red box bounds the scope of the framework.

7 FUTURE WORK

Next steps for this research project primarily consist of completing the framework development and validating the use of the method in a development process. The validation step concludes the thesis (Fig.2), this step constitutes instantiating the method in a development process for a training application. Using the VRTrain during the development phase, and evaluating the use of the framework against development without use of the framework.

REFERENCES

- [1] Steven Arild Wuyts Andersen, Peter Trier Mikkelsen, Lars Konge, Per Cayé-Thomasen, and Mads Sølvsten Sørensen. 2016. The effect of implementing cognitive load theory-based design principles in virtual reality simulation training of surgical skills: a randomized controlled trial. *Advances in Simulation* 1, 1 (2016), 20.
- [2] Mukesh Barange. 2015. Task-oriented communicative capabilities of agents in collaborative virtual environments for training.
- [3] Cédric Buche, Ronan Querrec, Pierre De Loor, and Pierre Chevaillier. 2004. Mascaret: A pedagogical multi-agent system for virtual environments for training. *International Journal of Distance Education Technologies (IJDET)* 2, 4 (2004), 41–61.
- [4] Pierre Chevaillier, Thanh-Hai Trinh, Mukesh Barange, Pierre De Loor, Frédéric Devillers, Julien Soler, and Ronan Querrec. 2012. Semantic modeling of virtual environments using mascaret. In *2012 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*. IEEE, 1–8.
- [5] Christiane Eichenberg. 2012. *Virtual reality in psychological, medical and pedagogical applications*. BoD–Books on Demand.
- [6] Vix Kemanji Ketoma, Philip Schäfer, and Gerrit Meixner. 2018. Development and evaluation of a virtual reality grocery shopping application using a multi-Kinect walking-in-place approach. In *International Conference on Intelligent Human Systems Integration*. Springer, 368–374.
- [7] Marius Koller, Philip Schäfer, Magdalena Sich, Julia Diemer, Mathias Müller, and Gerrit Meixner. 2018. Next Generation Virtual Reality Exposure Therapy Systems-A Study exploring Design Implications. In *2018 International Conference on Intelligent Systems (IS)*. IEEE, 528–535.
- [8] MI Sánchez, A de Antonio, and A de Amescua. 2005. SENDA: A Whole Process to Develop Virtual Environments. *Developing Future Interactive Systems, Idea Group* (2005).
- [9] Julien Saunier, Mukesh Barange, Bernard Blandin, Ronan Querrec, and Joanna Taoum. 2016. Designing adaptable virtual reality learning environments. In *Proceedings of the 2016 Virtual Reality International Conference*. 1–4.
- [10] José Luis Soler, Janaina Ferreira, Manuel Contero, and Mariano Alcañiz. 2017. The power of sight: using eye tracking to assess learning experience (LX) in virtual reality environments. *INTED2017 proceedings* (2017), 8684–8689.
- [11] Pieter Stappers and Elisa Giaccardi. 2017. Research through design. *The encyclopedia of human-computer interaction* (2017), 1–94.
- [12] Daniela Gorski Trevisan, Jean Vanderdonckt, and Benoît Macq. 2004. Conceptualising mixed spaces of interaction for designing continuous interaction. *Virtual Reality* 8, 2 (2004), 83–95.
- [13] Lode Vanacken, Chris Raymaekers, and Karin Coninx. 2007. Introducing semantic information during conceptual modelling of interaction for virtual environments. In *Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction*. 17–24.
- [14] Madis Vasser, Markus Kängsepp, Murad Magomedkerimov, Kälver Kilvits, Vladislav Stafinjak, Taavi Kivisik, Raul Vicente, and Jaan Aru. 2017. VREX: an open-source toolbox for creating 3D virtual reality experiments. *BMC psychology* 5, 1 (2017), 4.
- [15] Marvin V Zelkowitz and Dolores R Wallace. 1998. Experimental models for validating technology. *Computer* 31, 5 (1998), 23–31.

Crowd-powered Creativity Support Systems

Jonas Oppenlaender
jonas.oppenlaender@oulu.fi
University of Oulu
Oulu, Finland

ABSTRACT

Crowdsourcing has great potential in supporting humans to be more creative. This doctoral dissertation explores *crowd-powered creativity support systems* and covers a research arc from the fundamental prerequisites of leveraging crowds for creativity support to an accompanying set of case studies to clarify how complex creative work can be supported in practice.

CCS CONCEPTS

• **Human-centered computing** → *Interactive systems and tools*; • **Information systems** → *Crowdsourcing*.

KEYWORDS

creativity support systems, crowdsourcing, creativity

ACM Reference Format:

Jonas Oppenlaender. 2020. Crowd-powered Creativity Support Systems. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion)*, June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3393672.3398646>

1 INTRODUCTION

Crowdsourcing has been applied in a great number of applications, from finding cures for lower back pain [7] to supporting weight loss [8]. Since humans excel in divergent thinking, recombination, analogical transfer, and other fundamental characteristics needed for creativity, crowdsourcing has great potential in supporting creativity [10]. Supporting human creativity has been considered as one of the grand challenges of human-computer interaction [26].

And indeed, many studies and experiments in academia and industry routinely tap into the creative ability of the crowd, either for creativity-directed research or for simply eliciting ideas and creative artifacts from a large group of people [6]. However, the ability and willingness of the crowd

to participate in creative work and creative experiments is largely unquestioned by the academic community. Insights into the crowd workers' perspectives are rare, but important, as they may inform the design of studies with higher validity. It is imperative to develop an understanding of what crowd workers, as one of the primary stakeholder groups of crowdsourcing platforms, think about creative work. Such insights may inform the design of online systems and tools that rely on *crowd-powered creativity* [22, 24]. Much is to be learned about how the crowd perceives creative tasks and experiments, and what could be done to optimize such studies.

The research is motivated by the growing trend towards automation and a concern about the effect of artificial intelligence (AI) on the future work force. Creativity is what makes us human, and machines fail to epitomize creativity. Machines can interact with us in human-like ways, but they cannot tell whether a photograph is aesthetically pleasing, whether a painting will elicit feelings, or whether there is a deeper meaning behind a creative artifact. Creativity is thus one of the key skills that will secure human's place in future value chains. Research on human-centered AI [27], interactive machine learning [5] with a "human in the loop", human-machine-interaction [14], and hybrid AI [2] highlights the need of involvement of humans in future work flows, as a source of sentient human control in AI systems.

This research project explores the design space of *crowd-powered creativity support systems* [22] in two separate parts (see Figure 1). The first part (papers I–IV) investigates the crowd worker's perspective of creative work on crowdsourcing platforms. The findings from this part affect the requester's pre-ideation activities of outsourcing creative work (i.e., planning a crowdsourcing campaign, selecting the right crowdsourcing platform, and designing the task). The second part (papers V–VII) explores crowd-powered creativity support systems in three case studies.

2 RELATED WORK

Creativity. In this work, creativity is viewed as a domain-general concept. We follow the sociocultural view of creativity [1]. From this perspective, an idea or an artifact is considered creative if it is *novel* (or synonyms thereof, such as unique, original) and *appropriate* (or useful, effective) for a given domain, as judged by the domain's gatekeepers (the "field").

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7984-7/20/06.

<https://doi.org/10.1145/3393672.3398646>

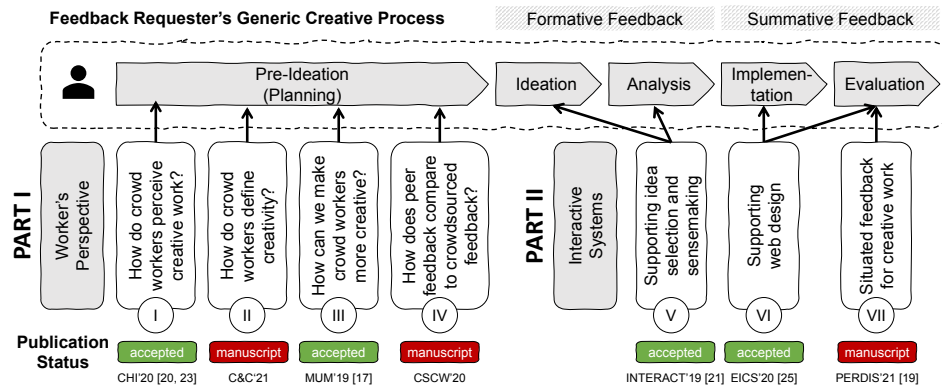


Figure 1: Structural overview of the research project.

Crowdsourcing. Crowdsourcing has become part of the toolkit of many researchers as a means for cost-effectively recruiting study participants and conducting experiments online. Our work focuses on investigating creativity on *general-purpose crowdsourcing platforms*. On these platforms, requesters publish tasks for an anonymous crowd to complete in exchange for a small monetary reward [9]. Other crowdsourcing platforms created for the specific purpose of eliciting creative works and ideas, such as Teefury, 99Designs, and many others, are out of scope of this work.

Crowd Feedback for Creative Work. Crowd feedback systems are computer-mediated systems that enable creative individuals to collect feedback and critique from a large number of people online [12, 15, 18, 30]. The present research project investigates the application of crowd feedback systems for supporting the creative process of the feedback requester. To this end, we adapt a generic model of the requester's creative process [31]: ideation (i.e., idea generation), analysis (e.g., idea selection and making sense of alternatives), implementation, and evaluation. In addition, we place emphasis on the early stages of the creative process, and extend the creative process model with pre-ideation activities, as found in the process typology in [6]. We acknowledge that this creative process is often iterative in practice and may evolve over time [3]. In this work, the generic process model serves as an overall framework along which we observe and investigate the crowd worker's perspectives in several creative activities.

3 CROWDSOURCING CREATIVITY SUPPORT

The remainder of this paper briefly presents completed studies (papers I and III) and systems (papers V and VI), followed by an outlook on future studies (papers II, IV and VII).

How do Crowd Workers perceive Creative Work? (I)

Resentment towards creative work and pre-exposure to common creative tasks could negatively impact the validity of

creativity studies and crowd-powered creativity support systems. In this work [20, 23], we explored what crowd workers think about creativity on two general-purpose crowdsourcing platforms: Amazon Mechanical Turk and Prolific. To the best of our knowledge, our work contributed the first worker-centered qualitative study of creative work on general-purpose crowdsourcing platforms.

The Study. We launched a survey task on the two crowdsourcing platforms. A mixed-method analysis of the workers' responses ($N = 215$) allowed us to develop complementary insights into creative work on the two platforms and a sense for different archetypal worker profiles.

Findings. Among other findings, we identified several archetypes of workers with different preferences and motivations for creative work. For instance, we found that professional crowd workers may have a negative attitude towards creative tasks. Different types of workers thus require different strategies for crowdsourcing creative work. Further, we found evidence that the overwhelming majority of workers (90%) in our sample preferred to work alone. Collaborative work flows have been researched in prior literature and some were found to be superior to non-collaborative work. Our finding stresses the need for careful design of the task and experiment, as enforced cooperation may cause issues in crowd-powered creativity support systems. Last, we found that some general-purpose crowdsourcing platforms may be more suitable for creative work than other platforms. The above findings highlight the need for novel mechanisms for recruiting workers interested in creative work.

Augmenting Creativity of Crowd Workers (III)

Microtask crowd work can be monotonous and repetitive and workers may not be "in the mood" for creative work. Prior research found that looking at a problem through the lens of a different person may lead individuals to be more creative [11, 13, 28]. In this work [17], we investigated the use of

roles in stimulating the divergent thinking of individuals in two complementary studies.

The Studies. In the first study, we implemented an online instrument for priming workers with roles and images, and recruited crowd workers ($N = 60$) to complete a divergent thinking task while assuming a role. In the second study, we observed the effect of roles on the ideation process of individuals when they reach an impasse in the flow of ideas.

Findings. In contrast to prior work, our analysis could not confirm computational priming having a significant effect on the outcome of a small batch of creative tasks. Our findings highlight that adopting roles can be an effective tool when creative individuals run out of ideas, but roles are not a silver bullet for improving divergent thinking.

4 CREATIVITY SUPPORT SYSTEMS

This section provides a brief overview of two crowd-powered creativity support systems created in this research project.

CrowdUI: Supporting Web Design (VI)

CrowdUI [25] is a crowd-powered creativity support system that enables the community of a website to visually provide feedback to the designer of the website, using the website itself as a canvas. Website visitors can directly modify (move, delete, resize) the elements of the web page. CrowdUI's multi-step process includes a tutorial to familiarize the user with the tool, as well as a peer review stage in which users rate other users' creations. The system allows designers to inspect the individual modifications of the user interface, and also provides further decision support by visually aggregating user modifications in heatmaps. The system was evaluated in a study with 45 users and 60 web developers with promising results.

GAS: Supporting Search and Exploration (V)

We designed and tested an intervention [21] to support writers in finding and exploring different ideas for a writing piece. The lightweight system draws on a crowdsourced database and supports complex searches for information, that is, searches with at least two criteria, through a faceted filtering interface. Our study compared the system to Google Search and found the system provides its users with support for exploring an information space and selecting ideas.

5 FUTURE WORK

The following three sections describe future work in the context of this research project.

How do crowd workers define creativity? (II)

We will provide an in-depth exploration into how crowd workers on general-purpose crowdsourcing platforms conceptualize creativity. A conceptual gap between how workers conceive creativity and how requesters of creative work (often

implicitly) define creativity may potentially impact the validity of creativity studies and creativity support systems. Ideally, this conceptual gap should be minimized.

Crowdsourced Feedback versus the Classroom (IV)

We will investigate the perceived usefulness and fairness of crowdsourced feedback in the context of an undergraduate design course. The study will add to the body of knowledge of applying crowd feedback systems in education [4, 29]. The aim of this study is to contextualize crowdsourced feedback and creativity support within a realistic design context.

SIMPLEX: Situated Feedback on Public Displays (VII)

While the willingness of situated workers to contribute to a task has been investigated (e.g., [16]), the willingness of a situated worker to provide summative feedback for creative artifacts has received little attention. With our third system, we will explore a situated system for supporting creative individuals with crowdsourced feedback for digital artifacts on public displays. The project aims to investigate the interplay and communication between feedback requesters and the situated crowd. In a preliminary needfinding study [19] with two artists and 12 users, we evaluated eight different types of feedback. We found a preference for simple feedback mechanisms on the side of the users, and identified the problem of making sense of many different feedback items as one of the key challenges for the requester of crowdsourced feedback.

6 CONCLUSION

Our work helps researchers and industry professionals who wish to harness the inherent convenience and power of crowdsourcing platforms in creativity-oriented studies and crowd-powered creativity support systems.

This research aims to advance our understanding of how complex and subjective creative work can be supported with a distributed non-expert crowd of anonymous workers. The work contributes towards defining crowd-powered creativity support systems as a research area within Human-Computer Interaction (HCI) and Engineering Interactive Computing Systems (EICS).

REFERENCES

- [1] Mihaly Csikszentmihalyi. 1996. *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Collins Publishers, New York, NY, USA.
- [2] Gianluca Demartini, Djellel Eddine Difallah, Ujwal Gadiraju, and Michele Catasta. 2017. An Introduction to Hybrid Human-Machine Information Systems. *Found. Trends Web Sci.* 7, 1 (2017), 1–87. <https://doi.org/10.1561/18000000025>
- [3] Kees Dorst and Nigel Cross. 2001. Creativity in the design process: co-evolution of problem–solution. *Design Studies* 22, 5 (2001), 425–437. [https://doi.org/10.1016/S0142-694X\(01\)00009-6](https://doi.org/10.1016/S0142-694X(01)00009-6)
- [4] Steven Dow, Elizabeth Gerber, and Audris Wong. 2013. A Pilot Study of Using Crowds in the Classroom. In *Proceedings of the SIGCHI Conference*

- on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 227–236. <https://doi.org/10.1145/2470654.2470686>
- [5] Jerry Alan Fails and Dan R. Olsen. 2003. Interactive Machine Learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (Miami, Florida, USA) (IUI '03). ACM, New York, NY, USA, 39–45. <https://doi.org/10.1145/604045.604056>
- [6] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the Landscape of Creativity Support Tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19). ACM, New York, NY, USA, Article 389, 18 pages. <https://doi.org/10.1145/3290605.3300619>
- [7] Simo Hosio, Jaro Karppinen, Niels van Berkel, Jonas Oppenlaender, and Jorge Goncalves. 2018. Mobile Decision Support and Data Provisioning for Low Back Pain. *Computer* 51, 8 (2018), 34–43. <https://doi.org/10.1109/MC.2018.3191250>
- [8] Simo Hosio, Niels van Berkel, Jonas Oppenlaender, and Jorge Goncalves. 2020. Crowdsourcing Personalized Weight Loss Diets. *Computer* 53, 1 (2020), 63–71. <https://doi.org/10.1109/MC.2019.2902542>
- [9] Jeff Howe. 2006. The Rise of Crowdsourcing. *Wired Magazine* 14, 6 (2006), 1–4.
- [10] Aniket Kittur. 2010. Crowdsourcing, Collaboration and Creativity. *XRDS* 17, 2 (2010), 22–26. <https://doi.org/10.1145/1869086.1869096>
- [11] Sheena Lewis, Mira Dontcheva, and Elizabeth Gerber. 2011. Affective Computational Priming and Creativity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 735–744. <https://doi.org/10.1145/1978942.1979048>
- [12] Kurt Luther, Jari-Lee Tolentino, Wei Wu, Amy Pavel, Brian P. Bailey, Maneesh Agrawala, Björn Hartmann, and Steven P. Dow. 2015. Structuring, Aggregating, and Evaluating Crowdsourced Design Critique. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (CSCW '15). ACM, New York, NY, USA, 473–485. <https://doi.org/10.1145/2675133.2675283>
- [13] Robert R. Morris, Mira Dontcheva, and Elizabeth M. Gerber. 2012. Priming for Better Performance in Microtask Crowdsourcing Environments. *IEEE Internet Computing* 16, 5 (2012), 13–19. <https://doi.org/10.1109/MIC.2012.68>
- [14] Donald A. Norman. 1994. How Might People Interact with Agents. *Commun. ACM* 37, 7 (1994), 68–71. <https://doi.org/10.1145/176789.176796>
- [15] Jonas Oppenlaender. 2019. Supporting Creative Workers with Crowdsourced Feedback. In *Proceedings of the 2019 ACM Conference on Creativity & Cognition* (C&C '19). ACM, New York, NY, USA, 646–652. <https://doi.org/10.1145/3325480.3326556>
- [16] Jonas Oppenlaender, Kennedy Opoku Asare, and Simo Hosio. 2018. CampusTracker: Assessing Mobile Workers' Momentary Willingness to Work on Paid Crowdsourcing Tasks. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (UbiComp '18). ACM, New York, NY, USA, 648–653. <https://doi.org/10.1145/3267305.3267550>
- [17] Jonas Oppenlaender and Simo Hosio. 2019. Design Recommendations for Augmenting Creative Tasks with Computational Priming. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia* (MUM '19). ACM, New York, NY, USA, Article 35, 13 pages. <https://doi.org/10.1145/3365610.3365621>
- [18] Jonas Oppenlaender and Simo Hosio. 2019. Supporting Creative Work with Crowd Feedback Systems. In *Proceedings of the DC²S² Workshop on Designing Crowd-powered Creativity Support Systems* (CHI '19). <http://arxiv.org/abs/2004.09204>
- [19] Jonas Oppenlaender and Simo Hosio. 2019. Towards Eliciting Feedback for Artworks on Public Displays. In *Proceedings of the 2019 ACM Conference on Creativity & Cognition* (C&C '19). ACM, New York, NY, USA, 562–569. <https://doi.org/10.1145/3325480.3326583>
- [20] Jonas Oppenlaender and Simo Hosio. 2020. What Do Crowd Workers Think About Creative Work?. In *Proceedings of the Workshop on Worker-Oriented Design: Expanding HCI Methods for Supporting Labor* (CHI '20). <http://arxiv.org/abs/2002.10887>
- [21] Jonas Oppenlaender, Elina Kuosmanen, Jorge Goncalves, and Simo Hosio. 2019. Search Support for Exploratory Writing. In *Human-Computer Interaction – INTERACT 2019* (LNCS 11748), David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris (Eds.). Springer, Cham, Switzerland, 314–336. https://doi.org/10.1007/978-3-030-29387-1_18
- [22] Jonas Oppenlaender, Maximilian Mackeprang, Abderrahmane Khayat, Maja Vukovic, Jorge Goncalves, and Simo Hosio. 2019. DC²S²: Designing Crowd-powered Creativity Support Systems. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI EA '19). Article W06, 8 pages. <https://doi.org/10.1145/3290607.3299027>
- [23] Jonas Oppenlaender, Kristy Milland, Aku Visuri, Panos Ipeirotis, and Simo Hosio. 2020. Creativity on Paid Crowdsourcing Platforms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20). ACM, New York, NY, USA, 1–14. <https://doi.org/10.1145/3313831.3376677>
- [24] Jonas Oppenlaender, Naghmi Shireen, Maximilian Mackeprang, Halil Erhan, Jorge Goncalves, and Simo Hosio. 2019. Crowd-powered Interfaces for Creative Design Thinking. In *Proceedings of the 2019 ACM Conference on Creativity and Cognition* (C&C '19). ACM, New York, NY, USA, 722–729. <https://doi.org/10.1145/3325480.3326553>
- [25] Jonas Oppenlaender, Thanassis Tiropanis, and Simo Hosio. 2020. CrowdUI: Supporting Web Design with the Crowd. *Proceedings of the ACM on Human-Computer Interaction* 4, EICS, Article 76 (2020), 28 pages. <https://doi.org/10.1145/3394978>
- [26] Ben Shneiderman. 2009. Creativity Support Tools: A Grand Challenge for HCI Researchers. In *Engineering the User Interface: From Research to Practice*, Miguel Redondo, Crescencio Bravo, and Manuel Ortega (Eds.). Springer, London, UK, 1–9. https://doi.org/10.1007/978-1-84800-136-7_1
- [27] Ben Shneiderman. 2020. Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy. *International Journal of Human-Computer Interaction* 36, 6 (2020), 495–504. <https://doi.org/10.1080/10447318.2020.1741118>
- [28] Jaime Teevan and Lisa Yu. 2017. Bringing the Wisdom of the Crowd to an Individual by Having the Individual Assume Different Roles. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition* (C&C '17). ACM, New York, NY, USA, 131–135. <https://doi.org/10.1145/3059454.3059467>
- [29] Helen Wauck, Yu-Chun (Grace) Yen, Wai-Tat Fu, Elizabeth Gerber, Steven P. Dow, and Brian P. Bailey. 2017. From in the Class or in the Wild? Peers Provide Better Design Feedback Than External Crowds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 5580–5591. <https://doi.org/10.1145/3025453.3025477>
- [30] Anbang Xu, Shih-Wen Huang, and Brian Bailey. 2014. Voyant: Generating Structured Feedback on Visual Designs Using a Crowd of Non-experts. In *Proc. Conf. Computer Supported Cooperative Work & Social Computing* (Baltimore, MD) (CSCW '14). ACM, New York, NY, 1433–1444. <https://doi.org/10.1145/2531602.2531604>
- [31] Liang Zeng, Robert W. Proctor, and Gavriel Salvendy. 2011. Can Traditional Divergent Thinking Tests be Trusted in Measuring and Predicting Real-world Creativity? *Creativity Research Journal* 23, 1 (2011), 24–37. <https://doi.org/10.1080/10400419.2011.545713>

Task Models Based Engineering of Interactive Systems

Célia Martinie

ICS, IRIT

Université Paul Sabatier Toulouse III

Toulouse, France

martinie@irit.fr

Philippe Palanque

ICS, IRIT

Université Paul Sabatier Toulouse III

Toulouse, France

palanque@irit.fr

ABSTRACT

This tutorial provides an overview and a practical experience on task models based engineering of interactive systems. It highlights how the recent advances in task modelling techniques can be exploited to design, develop and assess interactive systems. We present, from an engineering perspective and using examples from industrial case studies, how task models can support the design, development and evaluation of interactive systems. Furthermore, half of the tutorial focuses on the process of customizing task modelling notations for specific engineering needs and on its application using the tool supported HAMSTERS-XL notation.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Interactive Systems and Tools*

KEYWORDS

Interactive system engineering, task modelling, model-based design of interactive systems.

ACM Reference format:

Célia Martinie and Philippe Palanque. 2020. Task Models Based Engineering of Interactive Systems. In *Proceedings of ACM SIGCHI Conference on Engineering Interactive Computing Systems*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/1234567890>

1 Introduction

Task models is one of the very few means for explicitly and exhaustively describing user tasks. Task models can be used during several different stages of system design and development (e.g. user roles identification, system functions identification, user interface design, testing, training program design...) and by several stakeholders (e.g. human factor experts, system engineers, software engineers...)[7]. There are a lot of existing task modelling notations and each one matches a 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 third-party components of this work must be honored. For all other uses, contact the Owner/Author.

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-7984-7/20/06.

<https://doi.org/10.1145/3393672.3398647>

specific analysis objective (e.g. identification of knowledge required by the user, specification of the user interface dialog...) during the design and development of an interactive system [2]. Some of them are tool-supported and enable advanced engineering activities (e.g. generation of user interface software, partly-automated verification of consistency between software and user tasks...).

2 Aim and Objectives

This tutorial aims to provide an overview on how the recent advances in task modelling techniques can be exploited to design, develop and assess interactive systems. The main objectives are that, on completion of this tutorial, the participants will:

- Know the benefits of using tool supported task modeling techniques to design, develop and assess interactive systems and their user interfaces,
- Know at which stages of the design and development process and how task models can be used,
- Know how to identify the need for customizing a task modelling notation according to the objectives of the engineering activities,
- Know how to customize a task modelling notation to produce task models that fit the objectives of the engineering activities.

3 Intended Target Audience

This tutorial is open to researchers, practitioners, educators and students of all experience levels. The participants will be encouraged to bring a case study of an interactive system they had to or are engineering and for which they would like to model user tasks. We will guide them through the application of the task modelling notation customization for their specific engineering problem.

3 Basic Structure and Schedule

This tutorial will take place during a slot of 3 hours and will be composed of two parts. Each part will last 1h30 and will be separated from each other by the coffee break.

3.1 Part 1: Design, Development and Evaluation of Interactive Systems Based on Task Models (1h30)

In this part, we will present, from an interactive system engineering perspective, using examples from industrial case studies how task models can support:

- Design: identification of functions, temporal ordering of user actions, devices, required data, devices and objects ...
- Development: partly-automated verification of consistency between user tasks and software, dynamic generation of user interfaces...
- Evaluation: generation of scenarios, predictive assessment of task complexity, of efficiency...

3.2 Part 2: Tool Supported Customization of a Task Modelling Notation (1h30)

In this part, we will present stepwise guidelines to identify the need for customizing a task modelling notation according to the engineering objectives and according to the information that needs to be collected. We will also present the full tool based approach for customizing the HAMSTERS-XL notation and to produce customized task models. The participants will be guided during the application to specific examples or case studies of the guidelines and of the customization process. These examples or case studies can be brought by the participant or picked up in a selection of representative examples that we provide (comparison of interaction techniques for automated teller machines and for aircraft cockpits, description of possible user behaviors with an interactive system depending on taste stimuli...).

4 Teaching Method

One week before the tutorial, we will provide to the participants the HAMSTERS-XLE environment along with an installation procedure so that, for those that wish, their laptop is ready for the tutorial. The first part of the tutorial is an interactive presentation that encompasses three 10 minutes' slots for small hands-on exercises to familiarize with HAMSTERS-XLE environment and task models editing.

The second part presents the guidelines to identify the need for customizing a task modelling notation as well as the process for customizing a task modelling notation (approximately 20 minutes). This part then focuses on the practical application of the customization process (1h10) which is individually guided by the teachers.

5 Operational Requirements

This tutorial requires, in addition to the main video projector, a second video projector and associated display surface to guide the hands-on exercises while keeping displayed the main thread of slides.

6 Past Experience with the Tutorials

The instructors have several experiences in preparing and giving tutorials and courses on task model based design and assessment of interactive systems. The instructors taught a course on task analysis and task modeling exploiting HAMSTERS notation and tool at CHI 2015 [4], CHI 2016 [5] and INTERACT 2017 [6] (having at least 30 attendees each time). The instructors also taught a course on task model based assessment of effectiveness of Java applications at EICS 2018 [3] (having 10 attendees). One of the instructors prepared and give a recurrent course on “Design and Assessment of Systems Using Human Centered Approaches” at EUROCONTROL [1].

Compared to these previous tutorials, this proposal focuses on the benefits of using task models for engineering activities (software design and development, software verification) and on task modelling notation customization.

7 Resources

Each participant will be given pragmatic guidelines and the HAMSTERS-XLE task modeling environment for direct application to their engineering work. The HAMSTERS-XLE environment will be distributed to participants together with several examples going from toy examples to extracts from industrial projects. This set of materials (foundations, guidelines, tool and samples) is publicly available¹.

REFERENCES

- [1] Drogoul, F., Palanque, P. Design and Assessment of Systems Using Human Centered Approaches, EUROCONTROL, <https://trainingzone.eurocontrol.int/ilp/pages/coursedescription.jsf?catalogId=896431&courseId=7693921>, last accessed March 2020.
- [2] Martinie, C., Palanque, P., Bouzekri, E., Cockburn, A., Canny, C., Barboni, B. 2019. Analysing and Demonstrating Tool-Supported Customizable Task Notations. In Proc. of the ACM on Human-Computer Interaction, Vol. 3, EICS, Article 12 (June2019), 26pages.
- [3] Martinie, C., Navarre, D., Palanque, P. Exploiting Tasks Descriptions to Assess and Ensure Effectiveness during the programming of interactive Java software. Tutorial at EICS 2018, http://eics.acm.org/2018/program_tutorials.html
- [4] Palanque P., Martinie C. 2015. Designing and Assessing Interactive Systems Using Task Models. ACM CHI Extended Abstracts, 2465-2466.
- [5] Palanque P., Martinie C. Designing and Assessing Interactive Systems Using Task Models. 2016. ACM CHI Extended Abstracts, 976-979.
- [6] Palanque P., Martinie C., Winckler, M. Designing and Assessing Interactive Systems Using Task Models. In Proc. of INTERACT 2017, Part IV, Springer 2017.
- [7] Paterno, F. Task models in Interactive Software Sytems. 2001. Handbook of Software Engineering and Knowledge. World Scientific Publishing Co

¹ <https://www.irit.fr/recherches/ICS/software/hamsters/download.html>

ENTRAIN, Exploring New Territorial User Interfaces

Anne-Marie Dery-Pinna, Philippe
Renevier Gonin, Marco Winckler
Ecole Polytechnique Universitaire de
Sophia-Antipolis
Laboratoire I3S-CNRS-URA 1376
Sophia-Antipolis Cedex, France
firstname.lastname@univ-cotedazur.
fr

Christophe Kolski, Sophie
Lepreux
Univ. Polytechnique Hauts-de-France
LAMIH-UMR CNRS 8201
Valenciennes Cedex 9, France
firstname.lastname@uphf.fr

Jean Vanderdonckt
Université catholique de Louvain
(UCLouvain)
Louvain Research Institute in
Management and Organizations
(LouRIM Institute)
Louvain-la-Neuve, Belgium
jean.vanderdonckt@uclouvain.be

ABSTRACT

We aim at exploring the different approaches to design interactive applications for groups of users using a set of interacting surfaces to perform their tasks with an optimal user experience. Participants are invited to present both the models and/or design methods as well as the case studies and applications they are studying in this context. We would like to set up a discussion group in order to put each person's work in perspective with the notion of territoriality applied to ambient computing and multiple devices. The territoriality theory may serve as a basis for the design of complex interactive applications of quality. From these discussions will emerge a mapping of models and design methods that could be mutualized and combined.

CCS CONCEPTS

• **Human-centered computing** → *HCI design and evaluation methods; Graphical user interfaces*; • **Collaborative interaction**; Displays and imagers; • **Interactive systems and tools**; • **Software and its engineering** → *Distributed systems organizing principles*; • **Computing methodologies** → Distributed computing methodologies;

KEYWORDS

Design Space, Distributed User Interfaces, Interaction surface, Model-based approach, Multi-surface interaction, Territoriality, Territorial User Interfaces.

ACM Reference Format:

Anne-Marie Dery-Pinna, Philippe Renevier Gonin, Marco Winckler, Christophe Kolski, Sophie Lepreux, and Jean Vanderdonckt. 2020. ENTRAIN, Exploring New Territorial User Interfaces. In *ACM*

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

EICS '20 Companion, June 23–26, 2020, Sophia Antipolis, France

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7984-7/20/06.

<https://doi.org/10.1145/3393672.3398648>

SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '20 Companion), June 23–26, 2020, Sophia Antipolis, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3393672.3398648>

1 INTRODUCTION AND BACKGROUND

As interactive and connected devices become more and more varied [15], user interfaces are becoming more and more distributed [1, 13, 16]. Interactive applications therefore increasingly involve groups of users interacting with a set of interaction surfaces to carry out their tasks. The definition of design spaces, models and design methods have proven themselves in HCI design in the fields of plasticity, multi-modality, distributed interfaces. As in proxemic interaction [12], territorial user interfaces are governed by several parameters, such as location, partitioning, and orientation. The design methods most often exploit models and rules to propose a step-wise structured approach that leads to all or part of the interactive computer application. Such models assume to provide answers to the following questions:

- What are the specific data to an application domain (e.g., road traffic) and the constraints imposed on that domain?
- What are the tasks to be performed? What are the low-level actions (e.g., create, delete, modify, search) and the high-level tasks (e.g., putting a car on a road network)?
- Which users are involved in these tasks, depending on their roles (not fixed in time) in the groups?
- Which devices are used (e.g., from the smartphone to the large wall screen)?
- Which software platforms are used (e.g., what operating systems, what data exchange protocols are planned)?
- How is the physical environment in which the users are when performing their interactive tasks using the provided devices?

In order to design interactive applications dedicated to groups of users interacting with a set of interaction surfaces to carry out their tasks in complex environments, like museums, public places, offices, theatres, with a quality user

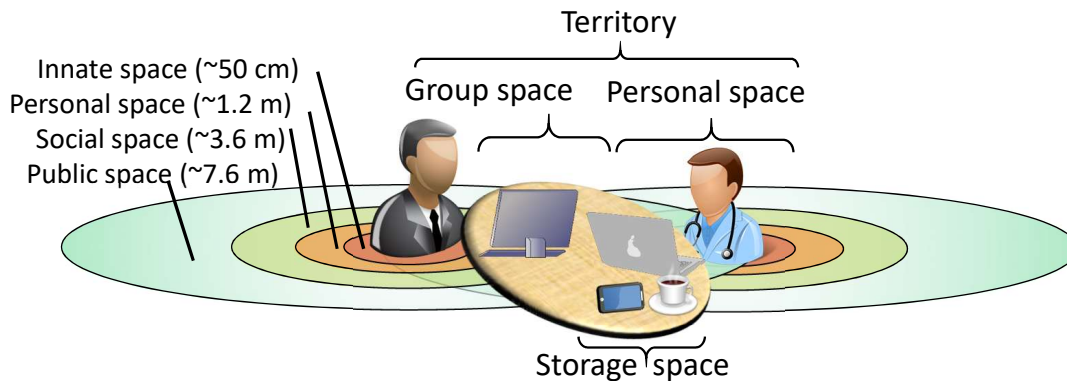


Figure 1: The concept of territoriality in UI design.

experience, we wish in this workshop to share the problems encountered and the solutions provided or envisaged. To this end, we wish to put submitted works in perspective with the notion of territoriality applied to ambient computing and multiple devices. The territoriality theory may serve as a basis for the design of complex interactive applications of quality.

Indeed the distribution of information is studied in the theory of territoriality for interactive tabletops [6–8, 16, 17]. The theory of territoriality [16] (See Figure 1) is indeed used to interpret the partitioning practices of the workspace constituted by the tabletop. Partitioning the workspace thus amounts to delimiting and managing different territories on the tabletop. Scott and Carpendale [16] distinguish three types of territory:

- (1) *Personal territories*, which are used to carry out autonomous activities/tasks (e.g. reading, writing and classifying resource items); they are also important in collaboration because they “provide a visible and accessible area for other members of the group, allowing them to follow the autonomous activities of a teammate”;
- (2) *Group territories*, which provide group members with a space to work on the product of the task and to help each other;
- (3) *Storage territories*, which allow to organize the resource elements on the table; they can be created on auxiliary surfaces that can be moved around the table.

A fourth type of territory is introduced by Li, Greenberg et Sharlin [11]: *private territories*, which are neither visible nor changeable publicly. Each territory combines three main characteristics [16]: (1) its purpose; (2) the typical activities or interactions it allows; (3) its spatial properties (size and shape). Territorial behavior is “based on perceived, attempted, or actual control of a definable physical space, object, or idea” [16]. The perception, partitioning, establishment, maintenance or change of territory (or transition between territories) are examples of territorial behavior.

Contributions from participants can help:

- To establish a literature review on the potential implications of the concept of territoriality in computer science.
- To make the link between the concept of territoriality and that of distributed interfaces,
- To list the models that could be used to design and develop the type of interactive applications targeted.
- To propose a method for developing interfaces by identifying input and output elements, guidelines and constraints to be respected.
- To identify case studies that can illustrate the problem and proposed solutions.

2 WORKSHOP OVERVIEW

Due to Covid-19 pandemic situation, the EICS 2020 Conference is canceled. So we reschedule the Workshop Organisation.

Workshop sessions dates

- Entrain, session 1 - initial contact: June, the 23th 2020: a remotely session
- Entrain, session 2 - thorough work: November, the 26th and the 27th 2020: at University Côte d’Azur (Nice, France), in face-to-face presentations if possible, remotely otherwise.

Workshop contributions

These two sessions are aimed towards gathering participants who are concerned by Models and Design Methods for Multi Surfaces User Interfaces especially excited about the opportunity to gather experiences around the territoriality notion.

Participants will be invited to submit an abstract on topics such as: the use of the concept of territoriality in the context of a distributed or non-distributed interactive application, the use of model definition in interactive applications for a group of users interacting with a set of interactive surfaces to perform their tasks, the use or definition of development

methods, rules, guidelines for interfaces on multi-surfaces, case studies to illustrate the problem and proposals for solutions in areas such as games, crisis management, collaborative design, trips in public places.

Workshop selection

- For the June session, we will select participants based on their positioning relative to the topic. Submit your positioning on the online form: <https://bit.ly/nomos-2020-06-23>
- For the November session, we will select up to 20 contributions. The workshop will begin with flash presentations. We will provide a consistent format for the presentations. We will discuss in sub-groups to put the work and the notion of territoriality into perspective. We will meet to synthesize the results of the groups as a mapping of each proposed models or approaches. We will establish a future research agenda.

Workshop website

The initiative web site is accessible at <https://nomos.i3s.unice.fr/fr/node/10>.

3 ORGANIZERS

The organizers of this Workshop are partners of the NOMOS project supported by the Hubert Curien Partnership Tourneol program which facilitates scientific cooperation between France and Belgium: Prof. M. Winckler, AM Dety Pinna and P. Renevier Gonin assistant professors in the I3S laboratory; S. Lepreux assistant professor and C. Kolski professor in the LAMIH laboratory UMR CNRS 8201, UPHF; J. Vanderdonckt professor in Université Catholique de Louvain (UCLouvain).

In ancient Greek, NOMOS (νόμος, “law” or “custom”) derives its etymology from the verb νόμομαι (to dispense or to allot), with NOMOS being the result of allotment or the manner of allotment or dispensing. We used this metaphor to express the concept of territorial user interface, a novel type of a distributed user interface that is regulated not by the physical principles of platforms or the implementation constraints imposed by widgets, but by the interpersonal space end users have among themselves.

The I3S team is basing part of its work on the composition of IU on model engineering [3–5]. The publications [10], [9] and [1] attest to the knowledge of LAMIH in terms of tangible interaction and multi-surface interfaces. The I3S and LAMIH laboratories explore the notion of territory in the design of distributed interfaces [5].

Jean Vanderdonckt is full professor in information systems leading the Louvain Interaction Lab, Université catholique de Louvain, Belgium. His research focuses on the contextual adaptation of user interfaces [14] and intelligent techniques

to support the adaptation of user interfaces at the time of design and/or execution [2]. He is a past president of the ACM IUI '04 conference and is co-chair of the ACM EICS '20 Full Papers conference.

4 FUTURE DIRECTIONS

The key outcomes of the workshop will be shared online, as a blog post on the workshop webpage. In addition, we also plan to organize a follow-up workshop in 2021 with a similar theme. This is aimed towards not only ensuring continual exchange and collaboration among the participants, but also to attract other members of the wider HCI community who might not be present at EICS.

A “white paper” synthesizing the results of the workshop can be considered if it is conclusive. In the longer term, a book in the HCI series could be considered, once sufficient contributions have been received.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support to this activity by the NOMOS (Nouvelle Organisation de Modèles Orientés Surface pour la conception de systèmes interactifs basés sur la territorialité), Hubert Curien programme (2020, first edition) and Wallonie Bruxelles International under convention PAD/CCA/SCU/AJO/ADD/FR/SORT/2019/545410.

REFERENCES

- [1] Amira Bouabid, Sophie Lepreux, and Christophe Kolski. 2019. Design and evaluation of distributed user interfaces between tangible tabletops. *Universal Access in the Information Society* 18, 4 (01 Nov 2019), 801–819. <https://doi.org/10.1007/s10209-017-0602-4>
- [2] Sarah Bouzit, Gaëlle Calvary, Joëlle Coutaz, Denis Chêne, Eric Petit, and Jean Vanderdonckt. 2017. The PDA-LPA design space for user interface adaptation. In *Proc. of the 11th IEEE Int. Conf. on Research Challenges in Information Science (RCIS '17)*. 353–364. <https://doi.org/10.1109/RCIS.2017.7956559>
- [3] Christian Brel, Philippe Renevier-Gonin, Alain Giboin, Michel Riveill, and Anne-Marie Dery. 2014. Reusing and Combining UI, Task and Software Component Models to Compose New Applications. In *Proc. of BCS-HCI '14*.
- [4] Anne-Marie Dery-Pinna, Jérémy Fierstone, and Emmanuel Picard. 2003. Component Model and Programming: A First Step to Manage Human Computer Interaction Adaptation. In *Human-Computer Interaction with Mobile Devices and Services*, Luca Chittaro (Ed.). Springer, Berlin, Heidelberg, 456–460.
- [5] Anne-Marie Dery-Pinna, Sophie Lepreux, Alain Giboin, and Philippe Renevier Gonin. 2019. Territories and Distributed HCI: Refining Rules and a Method for Designing Multi-Device Games. In *Proceedings of the 31st Conference on l'Interaction Homme-Machine: Adjunct (IHM '19)*. Association for Computing Machinery, New York, NY, USA, Article 8, 7 pages. <https://doi.org/10.1145/3366551.3370347>
- [6] Julian Edney. 1976. Human Territories: Comment on Functional Properties: Environment and Behavior. *Environment and Behavior* 8, 1 (1976), 31–47. <https://doi.org/10.1177/001391657600800109>
- [7] Mirko Fetter, David Bimamisa, and Tom Gross. 2016. Task-Based Focus and AdHoc-Focus-Territory: Novel Concepts for Shared Interactive Surfaces. In *Proceedings of the 2016 CHI Conference Extended Abstracts*

- on *Human Factors in Computing Systems (CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 1193–1200. <https://doi.org/10.1145/2851581.2892446>
- [8] Daniel Klinkhammer, Magdalena Mateescu, Carmen Zahn, and Harald Reiterer. 2018. Mine, Yours, Ours: Coordination through Workspace Arrangements and Territoriality in Tabletop Interaction. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia (MUM 2018)*. Association for Computing Machinery, New York, NY, USA, 171–182. <https://doi.org/10.1145/3282894.3282902>
 - [9] Sébastien Kubicki, Sophie Lepreux, and Christophe Kolski. 2012. RFID-driven situation awareness on TangiSense, a table interacting with tangible objects. *Personal and Ubiquitous Computing* 16, 8 (2012), 1079–1094.
 - [10] Sophie Lepreux and Jean Vanderdonckt. 2007. Towards A Support of User Interface Design By Composition Rules. In *Computer-Aided Design of User Interfaces V*, Gaëlle Calvary, Costin Pribeanu, Giuseppe Santucci, and Jean Vanderdonckt (Eds.). Springer, Dordrecht, 231–244.
 - [11] Jiannan Li, Saul Greenberg, and Ehud Sharlin. 2017. A two-sided collaborative transparent display supporting workspace awareness. *International Journal of Human-Computer Studies* 101 (2017), 23 – 44. <https://doi.org/10.1016/j.ijhcs.2017.01.003>
 - [12] Saul Marquardt, Nicolai Greenberg. 2015. Proxemic Interactions: From Theory to Practice. *Synthesis Lectures on Human-Centered Informatics* 8, 1 (2015), 1–199. <https://doi.org/10.2200/s00619ed1v01y201502hci025>
 - [13] Jérémie Melchior, Jean Vanderdonckt, and Peter Van Roy. 2011. A Model-Based Approach for Distributed User Interfaces. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '11)*. Association for Computing Machinery, New York, NY, USA, 11–20. <https://doi.org/10.1145/1996461.1996488>
 - [14] Vivian Genaro Motti and Jean Vanderdonckt. 2013. A computational framework for context-aware adaptation of user interfaces. In *Proc. of the 7th IEEE Int. Conf. on Research Challenges in Information Science (RCIS '13)*. 1–12. <https://doi.org/10.1109/RCIS.2013.6577709>
 - [15] Stephanie Santosa and Daniel Wigdor. 2013. A Field Study of Multi-Device Workflows in Distributed Workspaces. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. Association for Computing Machinery, New York, NY, USA, 63–72. <https://doi.org/10.1145/2493432.2493476>
 - [16] Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 294–303. <https://doi.org/10.1145/1031607.1031655>
 - [17] Jennifer Thom-Santelli. 2009. Expressing Territoriality in Collaborative Activity. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work (GROUP '09)*. ACM, New York, NY, USA, 389–390. <https://doi.org/10.1145/1531674.1531738>