

2ª Conferência Nacional em Interação Pessoa-Máquina

ACTAS

nteracção
2006

Universidade do Minho · Braga · 16-18 Outubro, 2006

Editores:

Teresa Chambel
Nuno Jardim Nunes
Teresa Romão
José Creissac Campos

Interacção 2006

Actas da 2a. Conferência Nacional em Interação Pessoa-Máquina

16, 17 e 18 de Outubro, 2006
Braga, Portugal

Organização



Universidade do Minho
Escola de Engenharia
Departamento de Informática



Apoio



<http://cesium.di.uminho.pt>



Patrocínios



Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR



UNIÃO EUROPEIA
FEDER



CCTC

CENTRO DE CIÊNCIAS E TECNOLOGIAS DA COMPUTAÇÃO

Grupo Português de Computação Gráfica

Mesa de Publicações
Rua Alves Redol, 9
1000-029 Lisboa
Portugal

<http://www.gpcg.pt>
email: pubs@gpcg.pt
Tel: + 351 21 3100300
Fax: + 351 21 3145843

Impressão: Copissaurio, Braga, Portugal.

Capa: Benedita Rangel / Logotipo: Teresa Chambel / Fotografia: José Creissac Campos

ISBN: 972-98464-7-2

Estas actas estão também publicadas em: <http://virtual.inesc.pt/interacao2006>

Prefácio

A concepção e desenvolvimento de sistemas interactivos envolve aspectos que vão desde a psicologia à engenharia de software. Recentemente verificou-se uma rápida evolução nesta área multidisciplinar, tanto a nível tecnológico, como do contexto de utilização da tecnologia. Assim, às tradicionais preocupações com a eficácia, eficiência e satisfação, juntam-se agora questões como entretenimento, design, estética, cooperação, ubiquidade e adaptabilidade. Paralelamente, o leque de utilizadores alvo dos sistemas é cada vez mais alargado, existindo a preocupação de desenvolver interfaces para pessoas e contextos de utilização com diferentes tipos de necessidades e requisitos ao nível físico ou cognitivo. Torna-se assim necessário perceber o impacto destas novas realidades no processo de interacção e no desenvolvimento de sistemas interactivos.

Depois do sucesso da primeira edição, organizada em Julho de 2004 na Faculdade de Ciência da Universidade de Lisboa, organizou-se agora a Interação 2006 – 2a. Conferência Nacional em Interação Pessoa-Máquina – numa iniciativa conjunta do Grupo Português de Computação Gráfica e do Departamento de Informática/Centro de Ciências da Computação da Universidade do Minho. Tal como na sua primeira edição, a Interação 2006 visou promover um ponto de encontro da comunidade interessada na Interação Pessoa-Máquina em Portugal. Reunindo investigadores, docentes e profissionais, permitiu a divulgação de trabalhos e a troca de experiências entre as comunidades académica e industrial.

Adoptando uma postura marcadamente multidisciplinar, a conferência desafiou os autores a abordar desde os fundamentos teóricos às práticas e aplicações emergentes, visando explorar sinergias nas seguintes dimensões:

- Aspectos Teóricos em Interação Pessoa-Máquina
- Concepção, Desenvolvimento e Avaliação de Sistemas Interactivos
- Tecnologias de Suporte à Interação
- Aplicações Interactivas

Em resposta a este desafio, foram recebidos trabalhos de Portugal, Espanha e Brasil.

Como resultado do processo de revisão dos trabalhos propostos, foram organizadas sete sessões técnicas em que foram apresentados 13 artigos longos e 16 artigos curtos, bem como uma sessão de posters e demonstrações em que foram apresentados 9 trabalhos. Para complementar o programa técnico foram ainda organizadas duas sessões proferidas por convidados internacionais de renome e uma sessão dedicada à indústria, em que se convidaram empresas e uma associação a apresentar a sua perspectiva sobre o tema da Usabilidade.

A conferência decorreu entre os dias 16 e 18 de Outubro de 2006, nas instalações da Universidade do Minho, Campus de Gualtar, em Braga. Procurou-se que ela fosse um retrato do trabalho que vem sendo desenvolvido nesta área em Portugal. Assim, as sessões técnicas desenvolveram-se em torno de sete temas principais, seleccionados a partir da análise dos trabalhos aceites: Acessibilidade, Psicologia Cognitiva e Colaboração, Concepção e Desenvolvimento, Interfaces Multimodais e Adaptativas, Realidade Virtual e Aumentada, Vizualização e Pesquisa de Informação, e Avaliação. Sem se tratar de uma classificação estanque, pensamos ser esta uma organização natural e abrangente dos trabalhos apresentados.

Terminamos agradecendo aos membros do Comité Organizador e da Comissão de Programa todo o seu trabalho, aos oradores convidados, Michael D. Harrison e Larry L. Constantine, bem como às instituições que aceitaram o desafio de participar na sessão dedicada à indústria (Associação Portuguesa de Profissionais de Usabilidade, MobiComp, Wintouch e Edigma.com) e a todos os autores. Agradecemos finalmente a todos os patrocinadores por nos ajudarem a tornar este evento possível. Esperamos que esta tenha sido mais uma contribuição para fomentar a partilha e colaboração da comunidade Interação Pessoa-Máquina em Portugal.

Braga, Outubro de 2006

Teresa Chambel
Nuno Jardim Nunes
Teresa Romão
José Creissac Campos

Presidente Honorário

António Dias de Figueiredo - UC

Comissão Organizadora

Co-Presidentes

José Creissac Campos - DI/UM; Adérito Marcos - DSI/UM

Financiamento e Ligação à Indústria

António Nestor Ribeiro - DI/UM; José Miguel Salles Dias - Microsoft e ADETTI/ISCTE

Sítio Web

Teresa Chambel - FCUL; José Creissac Campos - DI/UM

Organização Local

Flávio Ferreira - CESIUM, DI/UM

António Ramires Fernandes - DI/UM

Carlos Vilhena - GEMCC, DI/UM

Nuno Sousa - DI/UM

Comissão Científica

Co-Presidentes

Teresa Chambel - FCUL; Nuno Jardim Nunes - UMA

Posters e Demos

Teresa Romão - DI/UE

Comité de Programa

Ana Amélia A. Carvalho - DCTE/UM

Ana Paula Afonso - FCUL

Ana Paula Cláudio - FCUL

Anabela Simões - FMH

António Augusto de Sousa - FEUP

António Branco - FCUL

António Nestor Ribeiro - DI/UM

António Ramires Fernandes - DI/UM

Beatriz Carmo - FCUL

Beatriz Sousa Santos - IEETA/UA

Carlos Duarte - FCUL

Carlos J. Costa - ADETTI/ISCTE

Eduardo Dias - UE, YDreams

Francisco Rebelo - FMH

Isabel Trancoso - IST/UTL

Javier Rodeiro Iglesias - Univ. de Vigo, Espanha

João Álvaro Carvalho - DSI/UM

João Duarte Cunha - LNEC

João Falcão e Cunha - FEUP

João Madeiras Pereira - IST/UTL

João R. Caldas Pinto - IDMEC/IST/UTL

Joaquim Jorge - IST/UTL

Joaquim Madeira - DETI-IEETA/UA

Jorge A. Santos - DP/IEP/UM

Jorge Fernandes - Programa ACESSO/UMIC

Jorge Sousa Pinto - DI/UM

José Carlos Teixeira - UC

José Creissac Campos - DI/UM

José Miguel Salles Dias - Microsoft e ADETTI/ISCTE

Leonor Moniz Pereira - FMH

Lia Patrício - FEUP

Lola Dopico Aneiros - Univ. de Vigo, Espanha

Luís Carriço - FCUL

Luís Marcelino - FCT/UNL

Manuel Próspero dos Santos - FCT/UNL

Mário J. Silva - FCUL

Mário Rui Gomes - IST/UTL

Mário Vairinhos - UA

Nuno Correia - FCT/UNL

Nuno Guimarães - FCUL

Nuno Otero - Univ. of Hertfordshire, UK

Óscar Mealha - UA

Oscar Pastor - Univ. Politécnica de Valência, Espanha

Pedro Antunes - FCUL

Pedro Faria Lopes - DCTI-ISCTE/ADETTI

Pedro Rangel Henriques - DI/UM

Rafael Bidarra - Delft Univ. of Technology, Holanda

Raquel Santos - FMH

Rui José - DSI/UM

Silvia Abrahão - Univ. Politécnica de Valência, Espanha

Teresa Romão - UE

Vasco Branco - UA

Conteúdo

<u>ORADORES CONVIDADOS</u>	1
Ambient and mobile systems: new challenges for interactive systems analysis	
<i>Michael D. Harrison</i>	3
From Activity Theory to Design Practice	
<i>Larry L. Constantine</i>	5
<u>SESSÃO USABILIDADE</u>	7
Usabilidade - Técnicas de Baixo Custo	
<i>Associação Portuguesa de Profissionais de Usabilidade: Filipe Miguel Tavares</i>	9
A Decisiva Importância da Usabilidade nas Aplicações Móveis	
<i>MobiComp: Luís Carvalho</i>	11
Wintouch - Valor pela Intuitividade	
<i>Wintouch: Carlos Freitas</i>	13
DISPLAX - Desafios de Sistemas de Interação em Locais Públicos	
<i>Edigma.com: Rui Rodrigues</i>	15
<u>SESSÕES TÉCNICAS</u>	17
Acessibilidade	19
ATM Enhanced Accessibility for Disabled Persons: A Study of Time Limits	
<i>João Brisson Lopes, Leonor Moniz Pereira e António Raposo Cordeiro</i>	19
Avaliação do jogo de computador Boccia para adequação a uma equipa de jogadores com paralisia cerebral	
<i>Luís Carlos Bruno, João Brisson Lopes, Luís Filipe Garcia e João Madeiras Pereira</i>	27
O Sítio do Pico: um espaço de aprendizagem para crianças com paralisia cerebral	
<i>Anabela Cardoso da Silva, Elisabete Sousa Cunha, Pedro Rangel Henriques e Sandra Lopes</i>	35
Concepção de um Assistente de Software para o Desenvolvimento de Teclados para o Sistema Eugénio	
<i>Alexandra Moedas, Luís Garcia e Luís Bruno</i>	39
Psicologia Cognitiva e Colaboração	43
SeWiz: Personalizing Psychotherapy	
<i>Marco de Sá, Luís Carriço, Luís Duarte, David Cruz e Cátia Torres</i>	43
Supporting Group Analysis in Psychotherapy	
<i>Luís Carriço, Marco de Sá, Luís Duarte e Tiago Reis</i>	53
Dispositivos de Gestão da Atenção em Sistemas Colaborativos	
<i>António Ferreira e Pedro Antunes</i>	57
Online Communication in a Multimedia Course	
<i>Alexandra Rentróia-Bonito, André Martins, Tiago Guerreiro, Vitor Fernandes e Joaquim A. Jorge</i>	61
Concepção e Desenvolvimento	65
Engenharia Reversa de Sistemas Interactivos Desenvolvidos em Java2/Swing	
<i>João Carlos Silva, José Creissac Campos e João Saraiva</i>	65
Representación abstracta de interfaces de usuario DGAUI	
<i>Javier Rodeiro Iglesias e Susana Gómez-Carnero</i>	73

UML no Desenvolvimento de Sistemas Interactivos	
<i>José Creissac Campos e António Nestor Ribeiro</i>	77
Formalizing Markup Languages for User Interface	
<i>Luís G. M. Ferreira</i>	81
On the road to Rich Digital Books	
<i>Rui Lopes, Carlos Duarte, Hugo Simões e Luís Carriço</i>	85
Interfaces Multimodais e Adaptativas	89
Una Aproximación Sinestésica a la Interacción	
<i>Alberto José Garcia Ariza</i>	89
Gesture Human-Computer Interface for Command and Control	
<i>José Miguel Sales Dias, Pedro Nande, Nuno Barata e André Correia</i>	95
Avaliação de Interfaces Multimodais para Livros Falados Digitais com foco Não Visual	
<i>Carlos Duarte, Teresa Chambel, Hugo Simões, Luís Carriço, Eduardo Santos, Guilherme Francisco, Sérgio Neves, Ana Catarina Rua, João Robalo e Tiago Fernandes</i>	105
Controlo Miográfico de Dispositivos Móveis para Tetraplégicos	
<i>Tiago João Vieira Guerreiro e Joaquim A. Jorge</i>	115
JavaSketchIt2 – Using Relational Adjacency Grammars for Visual Syntax Parsing	
<i>Filipe M. Garcia Pereira, Manuel J. Fonseca e Joaquim A. Jorge</i>	119
Realidade Virtual e Aumentada	123
Engaging Virtual Agents	
<i>Ido A. Iurgel e Adérito F. Marcos</i>	123
Algoritmo de Visão para Humanos Virtuais	
<i>Pedro Semião, Maria Beatriz Carmo e Ana Paula Cláudio</i>	133
Faces – Biblioteca de Expressões Faciais	
<i>Janete Faustino, Ana Paula Cláudio e Maria Beatriz Carmo</i>	139
Usability in Virtual and Augmented Environments: A qualitative study	
<i>Angela Pimentel, Paulo Dias, Frank van Huussen, Pauline van der Horst, Jan-Willem Baggerman, Rafael Bidarra, Joaquim Madeira e Beatriz Sousa Santos</i>	143
Visualização e Pesquisa de Informação	147
Information Visualization Attributes and Operators for a Web Log Based Management System	
<i>Florin Zamfir, Óscar Mealha, Beatriz Sousa Santos e José Nunes</i>	147
Bridging the Gap Between Real and Electronic Documents	
<i>Renata Marin, Tiago Guerreiro, Daniel Gonçalves e Joaquim A. Jorge</i>	157
Pesquisa de Imagens na Web: Estudo de Usabilidade	
<i>Paulo Pombinho de Matos e Ana Paula Afonso</i>	167
Above and Beyond: Outer Space as a Metaphor for Document Browsing and Visualization	
<i>Daniel P. Gonçalves, Luís Bilo, Daniel Gonçalves, Manuel J. Fonseca e Joaquim A. Jorge</i>	177
Avaliação	181
IVY Trace Visualiser	
<i>Nuno Miguel Eira de Sousa e José Creissac Campos</i>	181
I-UsE: Integrated Usability Evaluation Environment	
<i>Luís Marcelino e Vasco Amaral</i>	191
Cognitive walkthroughs in the evaluation of user interfaces for children	
<i>Alexandre Mano e José Creissac Campos</i>	195
POSTERS E DEMOS	199
A System to Evaluate and Understand Routing Algorithms	
<i>Mario Marcelo Berón, Pedro Rangel Henriques e Maria João Varanda</i>	201
Conforto térmico humano: uma interface gráfica	
<i>Celina P. Leão, Pedro Arezes e S. F. C. F. Teixeira</i>	205
Desenho de um Sistema de Recolha de Estatísticas para o Boccia	
<i>Luís Alexandre, Gonçalo Fontes, Nelson Cavaco e Luís Garcia</i>	209

Desenvolvimento de uma Aplicação para a Visualização de Sessões de Hidrocinesioterapia	
<i>Elizabeth S. Carvalho e Adérito F. Marcos</i>	213
Estudos com Observadores para Avaliação de Qualidade: Influência do Contexto e do Tipo de Dados Avaliados	
<i>Samuel Silva, Carlos Ferreira, Joaquim Madeira e Beatriz Sousa Santos</i>	217
Evaluating an Activity-centered Ubiquitous Computing Framework for User Assistance in Public Spaces	
<i>Helder Pinto e Rui José</i>	221
Interior Design Tool for Common Users	
<i>Alfredo Ferreira, Roberto Medeiros, Rui Barreto, Manuel J. Fonseca e Joaquim A. Jorge</i>	225
Multi-Modal Painter: Pintar Usando as Mãos e Fala	
<i>Mariana Cerejo, José Santos, Alfredo Ferreira, Manuel J. Fonseca e Joaquim A. Jorge</i>	229
Plataforma Extensível para Interfaces Multimodais	
<i>Tiago Guerreiro, Vitor Fernandes, Bruno Araújo, Joaquim Jorge e João Pereira</i>	233
<u>ÍNDICE DE AUTORES</u>	237

ORADORES CONVIDADOS

Ambient and mobile systems: new challenges for interactive systems analysis*

Michael Harrison
Informatics Research Institute, Newcastle University, UK.

Abstract

Ambient and mobile systems bring potential for innovation to the way that information and services are offered to the users of physical environments such as leisure complexes, hospitals, airports and museums. Such systems are always on in the background, and can be used to deploy information to the user according to their context and location. Whether they are successful or not depends on how users experience the space in which they are situated as well as the more usual considerations of device and display usability and accessibility. What experiencing a system in a particular way might mean is difficult to express and implement in a system. For example, in an airport, creators of the environment might intend the ambient and mobile system to create for travellers an experience of place as opposed to a forbidding sterile space, or intend the system to allay the anxiety of travelling in an unfamiliar world. This talk is concerned with how to reason about whether systems will satisfy experience requirements and how this framework for reasoning might be integrated with more traditional analyses of interaction with the devices and displays involved. In particular it is concerned with whether existing techniques for analysing distributed systems might be applicable to these imprecise concepts that are so difficult to articulate. The talk is speculative about the set of tools and techniques required to achieve experience centred design. It explores a combination of formal techniques and prototyping to assess these further qualities of experience in the design of an interactive system. A focus on experience in ambient and mobile systems provides an opportunity for a fresh look at evaluation in interactive systems. Traditional notions of usability need reconsideration. Apart from the importance of experience to users, ambient and mobile systems have other distinctive characteristics that lead to a requirement for special treatment:

- *the impact of the environment as the major contributor in understanding how the system should work - its texture and complexity*
- *the use of location and other features in a software model of context to infer implicit or incidental user action - how natural and transparent this inference is.*

Methods for eliciting experience requirements are first discussed, exploring the limitations of scenarios and personae. Then the ingredients and requirements for a tool for combining the analysis of usability requirements with experience requirements are developed. Two examples are discussed. The first focuses on usability requirements for a mobile device to support process control and the second focuses on experience requirements by considering how information flows within an airport system. Finally a future agenda for completing a toolset to support the objectives is proposed.

Biography

Professor Michael Harrison is the Director of the Informatics Research Institute at the University of Newcastle-upon-Tyne, UK. He has worked both in universities (lectureship at Exeter and lectureship, senior lectureship and chair at York) and industry (software designer with CAP and project manager at Inmos). His research has reflected this combination of backgrounds, being concerned with the application of formal methods to the problem of user interface design. He has recently moved substantially away from the use of formal notations to formal methods that can integrate with industrially relevant processes and approaches. His research is now focussed on ambient mobile intelligence, extending analysis techniques to systems that will dominate our future. In this regard he works within an interdisciplinary framework.

*This keynote was supported by the IVY Project, funded by The Portuguese Science and Technology Foundation (FCT),

and The European Regional Development Fund (FEDER), under contract POSC/EIA/56646/2004.

From Activity Theory to Design Practice*

Larry Constantine
Constantine & Lockwood, Ltd.

Abstract

What is the proper focus of interaction design? Is it more about people or about the activities in which they engage? Psychologist and usability guru Donald Norman has argued that to create better tools that support user performance, designers need to turn to activity-centered approaches grounded in activity theory. To put activity theory into practice, an appealing but loose conceptual framework needs to be transformed into a systematic discipline. Activity modeling is a disciplined approach that focuses on the simple representation and organization of those elements of human activity that are most relevant to informing and guiding an activity-centered design process. This keynote will introduce activity modeling and show how it fits in the larger visual and interaction design process.

Biography

Larry Constantine, IDSA, is an internationally recognized leader in design methodology and product usability. He is an award-winning designer specializing in interaction design and techniques for enhancing user performance. A persistent innovator with patents in human-machine interaction to his credit, he is co-inventor of essential use cases, regarded by many as a best practice in user requirements definition and task modeling. With Lucy Lockwood, he developed usage-centered design, the widely practiced model-driven process based on essential use cases. One of the pioneers of modern software engineering theory and practice, Constantine is the originator of many widely used models and techniques, including dataflow diagrams and the quality metrics of coupling and cohesion. His notational innovations have been incorporated into modern object modeling techniques, including the Unified Modeling Language.

*An award-winning author, he has published more than 175 papers and 17 books in both the human sciences and information sciences, including the classic text, *Structured Design*, written with Ed Yourdon, and the award-winning *Software for Use*, written with Lucy Lockwood. His papers have been widely reprinted and his books have been translated into nine languages. In wide demand as a presenter and teacher, he has taught in 18 countries around the world and has keynoted numerous international conferences. His clients have included such major international technology leaders as McKesson, Siemens AG, Pitney-Bowes, DaimlerChrysler, Lockheed, Nortel Networks, IBM, CAP Debis, SAP, Dow Jones, and Canon Information Systems.*

*This keynote was partially supported by Microsoft Portugal.

SESSÃO USABILIDADE

Usabilidade – Técnicas de Baixo Custo

Filipe Miguel Tavares
Associação Portuguesa de Profissionais de Usabilidade

Resumo

A Usabilidade, enquanto opção estratégica da empresa, maximiza o retorno do investimento sustentando e incrementando a comunicação com o cliente/utilizador. Nesta perspectiva, é de importância crucial a sua integração no desenvolvimento e nos respectivos processos de pensamento e de gestão. Assiste-se contudo a um certo pudor na assunção deste “novo pensamento”, sustentado essencialmente por uma ideia latente de que a Usabilidade acarreta um grande custo orçamental.

Esta apresentação visa dotar empresas e equipas de projecto de técnicas de Usabilidade de baixo custo, pensadas para que constituam uma base ideológica, integráveis no pensamento consciente e sub-consciente das equipas e respectivos profissionais.

Sobre a APPU

A APPU - Associação Portuguesa de Profissionais de Usabilidade é uma associação profissional sem fins lucrativos que procura representar os profissionais de Usabilidade portugueses, bem como divulgar os benefícios da Usabilidade às empresas e aos cidadãos. A APPU, nascida de um grupo informal formado em 2001 por profissionais do sector, foi formalizada em Julho de 2005, contando actualmente com cerca de 50 membros profissionais do sector.

A Decisiva Importância da Usabilidade nas Aplicações Móveis

Luís Carvalho
MobiComp

Resumo

As aplicações móveis constituem casos particulares de aplicações nos quais o binómio usabilidade-utilidade é decisivo. É necessário que a aplicação por mais útil que seja, disponha dos mecanismos que permitam aos seus utilizadores a invocação fácil das funcionalidades e uma interacção guiada por metáforas entendidas por estes. O desafio é ainda maior quando aliado ao carácter ubíquo das aplicações, os ecrans e mecanismos de manipulação apresentam diferenças substanciais em relação às aplicações desktop.

Esta apresentação demonstra o esforço que a MobiComp devota a esta temática apresentando dois casos de estudo. O primeiro aborda a construção de um novo componente de interacção por forma a promover um maior aproveitamento da área disponível para apresentação de conteúdos, permitindo um acesso fácil às funcionalidades existentes. O segundo exemplo, o MobileKeeper, aborda a construção de uma aplicação em que uma análise das tarefas que o utilizador executa, altera de forma significativa a interface com o utilizador, ao se privilegiar as tarefas que o utilizador necessita em vez de se optar por uma abordagem orientada ao agrupamento de funcionalidades.

MobiComp

Established in 2000, MobiComp is a leading provider of mobile solutions in the area of personal information management, content sharing, and service promotion. The company's suite of products allows mobile operators and service providers to boost data revenues and reduce churn by capitalising on end user generated content.

MobiComp's solutions have been taken into use by mobile operators, service providers, and select enterprises across the world, including Vodafone, TMN, Optimus, Esprit, and MilleniumBPC.

Headquartered in Braga, Portugal, MobiComp currently has offices in London, Madrid, Helsinki, Dubai and Lisbon. For more information on MobiComp, please visit www.mobicmp.com.

Wintouch - Valor pela intuitividade

Carlos Freitas
Wintouch

Resumo

Esta apresentação versa sobre os cuidados a ter no desenvolvimento de soluções informáticas a serem utilizadas por organizações cuja força de trabalho seja constituída por colaboradores com baixa formação e com grande rotatividade.

Wintouch

Wintouch, é uma empresa 100% nacional sediada em Braga, especializada no desenvolvimento de software para os segmentos de mercado da Restauração, Hotelaria, Retalho e Saúde. Com os seus 7 anos de existência, é considerada um dos principais players do nosso mercado, dada a qualidade do seu software, nomeadamente a sua usabilidade. Este êxito tem sido transposto para outros países, nomeadamente Espanha, onde constam da sua lista de clientes cadeias de referência como Burger King, KFC, Pans and Company, Pizza Hut, Bocatta, entre outros.

DISPLAX - Desafios de Sistemas de Interação em Locais Públicos

Rui Rodrigues
Edigma.com

Resumo

O mercado evolui constantemente ao nível da exigência e das expectativas de desempenho de produtos e serviços. Esta exigência, aliada à concorrência, obriga que as empresas de uma forma geral procurem diferenciar-se não só pelos produtos e serviços que oferecem, mas também pelas estratégias de marketing, apostando em acções arrojadas e inovadoras na forma de atingir o público alvo e preferencialmente interagindo com ele. As equipas de marketing, design e publicidade têm assim um papel fulcral na interligação dos agentes activos do mercado. Os criadores de design e tecnologia entram neste processo para auxiliar os fornecedores e marketeers a alcançar os seus objectivos, através da criação e implementação de sistemas inovadores que aliem o design funcional, os conteúdos e as metáforas de interacção por eles potenciadas.

A EDIGMA.COM SA, como empresa criadora de tecnologia e design, tem apostado na investigação e desenvolvimento na área dos sistemas e conteúdos interactivos para instalação em locais públicos. A criação destes sistemas implica não só componentes de hardware e design, mas também a idealização e adaptação de diferentes formas de interacção que surpreendam. Esta interacção contempla pelo menos dois níveis: a interacção público-sistema e a interacção gestor-sistema. Esta apresentação foca dois tipos de sistemas interactivos desenvolvidos pela EDIGMA.COM SA – Displax Interactive Holo (montra interactiva holográfica) e Displax Interactive Floor (piso interactivo de grande dimensão) – que apresentam constantes desafios ao nível de implementação física e tecnológica e ao nível dos modelos de interacção proporcionados pelos diferentes contextos.

Edigma.com

Edigma.com, SA - Technology, Contents and Consulting. Fundada em 2000, a EDIGMA.COM produz tecnologias interactivas, sistemas de ecrãs opacos e transparentes de grande dimensão sensíveis ao toque através de vidro e janelas, sistemas de gestão de conteúdos e aplicações multimédia e Internet. Actualmente existe um programa de I&D a ser desenvolvido em parceria com a Universidade do Minho e o CCG/ZGDV. Em 2004 o produto Displax®-Interactive Window foi lançado com grande sucesso. O Displax começou como tecnologia altamente inovadora que potencia a criação de conteúdos interactivos em ecrãs holográficos que surpreendam e cativem o público; neste momento, é uma vasta gama de diferentes sistemas interactivos para locais públicos, incluindo pisos interactivos, e está continuamente a ser alargada.

SESSÕES TÉCNICAS

ATM Enhanced Accessibility for Disabled Persons: A Study of Time Limits

João Brisson Lopes
Dep. Eng^a. Informática,
IST/UTL, Lisboa
brisson@ist.utl.pt

Leonor Moniz Pereira
Dep. Ed. Especial e Reabilitação,
FMH/UTL, Lisboa
lmpereira@fmh.utl.pt

António Raposo Cordeiro
Dep. De Design,
FA/UTL, Lisboa
araposocordeiro@netcabo.pt

Abstract

This paper presents an accessibility study carried out to determine the most appropriate timeout values that allow users with disabilities to use ATM terminals to carry out banking operations. The paper describes the methodology applied to the study which consisted of a questionnaire phase, followed by a test phase which took place in an ATM test bed from which recommendations were drawn. This study addressed the diversity of problems presented by users with visual disabilities (low vision and blindness), motor disabilities (wheelchair, assistive walking devices and limited dexterity) and intellectual disabilities.

Keywords

Accessibility, user centred design, usability, experimental studies, ATM, timeouts.

1. FOREWORD

The most common complaint coming from ATM users is that the time limits to complete many operations are rather short. In fact, these limits, known as timeouts, were originally set with the following in mind:

- Safety – timeouts protect users against events such as forgetting to take out the card or the money or when the user is at loss and does not press the cancel button.
- Convenience – ATMs assume appropriate default choices to yes/no-like dialogues when timeouts expire.
- Turnaround time – the knowledge that timeouts are in effect prevents users from delaying, making ATMs available for more users and more banking operations.
- Security – timeouts protect users from long exposure and malicious intents and interference.

Current timeout values assume that users know exactly which transaction (operation) they want to make, how to navigate menus to reach the transaction dialogue and that the input data for the transaction is ready and present (e.g., a slip of paper with an account full number).

But it happens that many times users hesitate because they are uncertain of how to reach a given transaction dialogue. Other times they take too long to make sure that the transaction data has been correctly typed. In both cases the time limit expires and the user has to start all over again.

Furthermore, it is assumed that no environmental problems hamper users on their tasks. This is often not the case since ATM terminals may be located either too high or too low, in very narrow spaces or exhibit the very recurring and annoying screen glare due to direct bright sunlight in the daytime or improper, perhaps incorrect, illumination at night time.

There is no any way for ATM software to detect any of the above. However, users need more time than usual to carry out an operation and, therefore, it is more likely that the time allowed for the operation will expire.

1.1 Timeouts and Disabled Persons

These problems are even further compounded for disabled persons. Such users tend to be slower than the average. Moreover, the knowledge that time limits are short is yet another stress factor making disabled persons hurry up, thus creating the right conditions for them to make mistakes. Recurring mistakes and failures have the effect of discouraging everybody (and not only disabled persons) from using ATMs.

In a general way, ATM operators and designers overlook such user problems. Also, research on these matters is scarce, albeit inexistent, because most researchers tend to address the conditions for physical accessibility to ATMs and not their operation.

This reflects in the existing standards like CEN 15291 [CEN15291] that limit themselves to the enunciation of vague and general principles for accessibility in interface design. Another example, the Brazilian standard for ATM accessibility [ABNT15250], recommends only that there

should be a 2s pause between audio messages and that users should be allowed a 5s wait time when prompted to start data input.

Other studies, like Davies et al [DLW02], concentrate on the rehabilitation of disabled persons. These studies focus on making such persons use ATMs again and not on ATM design issues.

This paper addresses the experimental determination of adequate timeout values for the different disabilities presented by disabled persons and, therefore, how to increase ATM accessibility

It is important to stress that ATM enhanced accessibility for elderly or disabled persons benefits not only such persons but the generality of ATM users.

This work was part of a larger study on how to improve accessibility to ATM terminals for elderly and disabled persons [UTL06] carried out under a contract from SIBS (Sociedade Interbancária de Serviços) by the Technical University of Lisbon. The study involved the Department of Design of the Faculty of Architecture, the Computer Science Department of Instituto Superior Técnico and the Special Education and Rehabilitation Department of the Faculty of Human Kinetics.

2. TARGET USERS AND USERS GROUPS

The designation “disabled persons” encompasses such a very large number of disabilities that the current study, in its first step, opted to consider only three broad kinds of disabilities:

- Visual impairment (blindness and low vision).
- Motor impairment (wheelchair, device assisted walking, limited dexterity).
- Mental impairment.

Furthermore, it is well worth noting that a person will often exhibit more than one typified disability.

No deaf persons were targeted in this study because deafness alone is no impairment to ATM use since deafness does not affect vision or motor dexterity, the two major requirements to operate an ATM terminal successfully without timeouts.

Disability/ Age Group	Total	Details
Visual	74	44 blind 30 low vision
Motor	74	30 wheelchair 37 device assisted walking 7 limited dexterity
Mental	118	
Elderly persons	80	

Table 1: Persons who answered the questionnaire according to disability type or age group.

3. METHODOLOGY

This study was conducted in three phases [Pereira07]:

- Identification of the ATM models in operation
- User survey
- Trials involving users

The first phase was carried out in order to identify the different ATM terminals in operation in Portugal because of their diversity of device layout and keypad layout, to name just a few.

In the second phase a questionnaire was prepared and distributed to associations of disabled persons, day centres, clubs and other institutions frequented by disabled and elderly persons. Questionnaires were filled in by users or by staff of this study if the person would not be able to do it.

In all 346 persons answered the questionnaire. Table 1 shows the number of answers received according to type of disability.

Of these 206 were ATM users while the remaining 150 were not.

Processing of the questionnaire answers showed some interesting results. One of the most relevant was that the short time allowed to carry out operations. A related conclusion was that respondents not using ATMs or who had quit using them would consider using ATMs if they were given the chance to train using ATMs without time limits, especially in the case of the more complex tasks.

The questionnaire phase also allowed us to identify users who could be retained for third phase, the trial phase.

The third phase aimed at testing solutions for enhanced accessibility on a test bed where it was possible to change its physical layout, illumination and software.

The software was runtime configurable, i.e., had many alternative options that could be set on and off while tests were running.

The software also had data collection built in so that it was possible to collect user data.

The 20 users who participated in the third phase were chosen so that they each one presented at least one of the following conditions:

- Difficulties pressing a key and introducing the card in the slot
- Wheelchair or walking assistive device usage
- Height less than 1.35 m or higher than 1.85 m
- Ability to read small sentences or simple numbers or “read” them from audio output devices
- Ability to follow instructions

3.1 Tasks Involved

It is well worth noting that disabled persons that operate ATMs and experiment timeout operation termination are usually faced with difficulties in at least one of the following [Gill98]:

1. Insert the card
2. Read the screen
3. Use of the keypad, screen lateral keys or a touch screen
4. Listen to audio output
5. Retrieve the card

These tasks are essential to complete a transaction at an ATM terminal. Any delays while performing any of them add to the overall time taken and increase the likelihood of a timeout happening.

However, there can be simple and complex transactions. In Portugal, the most frequent operations are money withdrawal, account balance inquiries, prepaid mobile telephone payments and paying bills. Besides having to select and carry out these operations by themselves, users must at all times insert the card, key in the PIN code and take out the card and any bills and receipt slips.



Figure 1: A user at the ATM test bed.

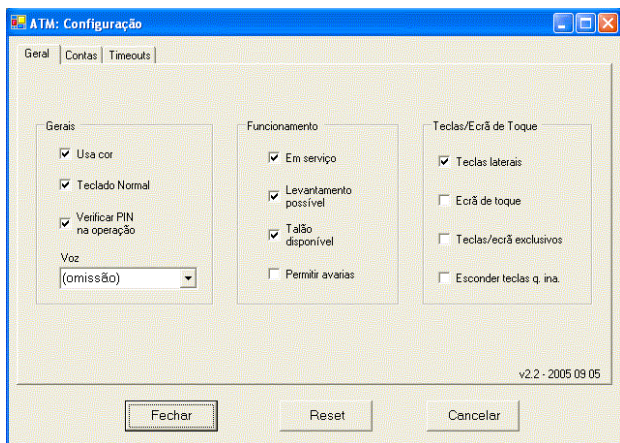


Figure 2: Main ATM test bed configuration panel.

Although the test bed software was capable of simulating all the banking operations available from all SIBS network ATM terminals, this study concentrated in the following most frequent and relevant operations:

- Cash withdrawal
- Payment of bills
- Prepayment of mobile telephones

3.2 Trials and the Test Bed

Trial users were asked to perform the operations listed above on an ATM test bed specifically designed to test accessibility improving solutions. Figure 1 shows a test user performing tasks at the test bed.

Because lighting conditions (and glare) affect user performance (and success), it was possible to change lighting conditions. Light sources located on the sides and above the test bed provided different lighting conditions. The room where the trials took place also allowed the control of other environmental conditions as well as illumination.

The test bed software was specifically designed to reproduce the software that was currently installed in all ATMs of the SIBS network. However, its software behaviour could be modified through a configuration panel that allowed the setting of many options through a series of configuration panels. Figure 2 shows the main configuration panel. The timeouts configuration panel (see Figure 3) allowed several ways to control timeouts.

Timeouts could be individually configured. This meant that the time limit to input the PIN code, the time limit to select an operation and the time limit to fill in the data to a bill could be individually set even a runtime, for instance. This allowed the exploration of values not tied to the values that were in effect in ATMs from the SIBS network at the time.

The software also allowed adapting almost all individual time limits to the pace of input of a relevant data item, e.g., the PIN code. Furthermore, all time limits could be multiplied by a given factor. In practice, we could set this factor so high that no time limit would be reached.

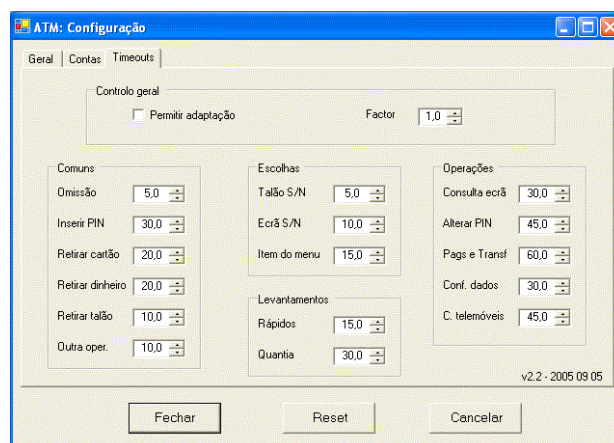


Figure 3: ATM test bed timeouts configuration panel.

The test bed software was also able to collect usage data, namely user actions. Particularly, the software recorded which keys the user pressed and the time at which each key was pressed on a log file.

Since the log file also contained the times of screen transitions, it was therefore possible to calculate the time taken by the user to complete a dialogue, make a selection on a menu and the user's pauses and hesitations.

All this data was automatically recorded on the log file and was complemented by notes taken by an experimenter who would observe the user performance (effort, insecurity, remarks, etc.).

After a briefing interview designed to identify the user's main concerns, priorities and needs, the user would then get acquainted with the test bed and freely explore it. This would be followed by the execution of the transactions assigned to the user by the experimental protocol.

After finishing the transactions, a final debriefing interview would follow to evaluate accessibility and other relevant issues from the user's point of view.

4. TRIAL RESULTS

The following presents the results obtained from trials that involved the following kinds of users:

- Blind
- Motor impaired
- Mentally impaired

No elderly persons or persons with low vision were involved in these trials because the disabilities they exhibit are encompassed by the disabilities of the users listed above.

4.1 Blind Users

All vision impaired users selected for the trials were blind users since they constitute the most demanding user subgroup in the visually impaired category. The reason for this choice was that we wanted to assess the operation of ATMs by blind persons and determine the appropriate values for timeouts when using the ATM interface variant that provides auditory output.

SIBS specifically designed this interface for this kind of users a few years ago, but no specific study of timeouts was done at the time. Instead, the timeout values for the average user were adopted for this interface without considering the specific needs of blind users.

An already large number of SIBS network ATM terminals present this special interface with different menu and dialogue screens and sound output.

ATM terminals enter this special interface operation mode if the user presses keypad key 5 right after keying in their PIN code.

In this audio output mode, ATM terminals read out aloud the choices and instructions through loudspeakers. Some recent terminal models have also a 3.5 mm audio earphone jack that provides privacy and increased security.

The audio operation mode restricts the operations the user can make to the following¹:

- Cash withdrawal
- PIN code change
- Bill payment

Because blind users mentioned that they considered very important to be able to make mobile phone prepayments, we decided to add this operation to the list above for exploratory purposes and for determining its appropriate timeout values. It must be stressed that this operation is not currently available in audio output operation mode.

The operation of an ATM terminal in audio output mode is as follows. After pressing key 5, the user is prompted to select one of the operations listed above and then taken to the normal screen dialogue for the operation chosen. The terminal reads out any instructions specific to the operation that was selected and, as it is the case of cash withdrawal, restricts the dialogue to the one where the user types in the amount to withdraw.

Depending on the operation selected, the terminal may present the user with a confirmation dialogue right after the operation dialogue. Confirmation dialogues are not usually entered in normal mode (i.e., without audio output) but are used by SIBS ATMs in most of the operations carried out with audio output.

In order to be able to determine a most comprehensive set of timeout values to recommend, we decided to present confirmation dialogues for all operations carried out in audio output mode. This choice was based on the opinions collected from the focus group of blind persons at the time the test bed was validated.

Confirmation dialogues read out the values keyed in by the user (e.g., amount to withdraw) and then prompt the user to confirm (green key) or to cancel (red key) the operation.

Another difference of the audio output mode relative to normal operating mode is that the user has no choice to request or reject printing a paper slip with the details of the operation. In audio output operating mode this receipt is always printed.

For these trials, as well as for the trials with other kinds of disabled persons, the timeouts built into the test bed software were completely disabled. This allowed the determination of the exact times taken by the users without masking out those times above the current SIBS timeout values and, therefore, provided data that could be used to set new timeout values ranges appropriated for each kind of disabled users.

¹ Recently, another operation, MB purse card fill-in was removed following the end of this type of card.

Task	Min.	Max.	
PIN code input	Overall	5	32
	Code input	3	22
Key 5	1	9	
Audio menu selection	7	24	
Card withdrawal	No cash	3	46
	Cash	10	28
Cash withdrawal	Overall	8	25
	Amount selection	4	12
	Confirmation	5	14
Mobile phone prepayment	Overall	46	58
	Data input	39	49
	Confirmation	17	28
Bill payment	Overall	43	45
	Data input	36	37
	Confirmation	7	25

Table 2: Minimum and maximum times (in seconds) taken by blind users at the ATM test bed.

Table 2 shows the maximum and minimum values that were recorded in the trials with blind persons.

The values ranges are relatively wide because subjects' ATM usage expertise varied widely. Some users were experienced in using keyboards and keypads with a relatively low number using them everyday. On the other end of the spectrum, there are users who only sporadically use them and this shows in the higher values that were recorded.

However, the most important values are represented by the maximum values that can be read as "no (much) lower than this value".

The overall values shown on the table represent the time taken from the moment the audio prompt starts to the moment the user finishes typing the input data. The difference between an overall time and its corresponding data input time (taken from the moment the user presses the first key) should roughly represent the time taken by the software to output the audio instructions plus the user to reaction time.

This reflects in the minimum and maximum values quoted for mobile phone prepayments and the payment of bills but not in cash withdrawal operations. This is because this operation is the most frequent operation and frequent users need no longer listen to the audio output. Instead they can immediately start typing in the amount they wish to withdraw² because they know or decided upon this

² Audio output stops when the user presses the first key.

amount in advance. Infrequent users still need some time to make up their mind about how much to withdraw.

4.2 Motor Impaired Users

The tests with motor impaired users were conducted using the normal user interface (no audio output) but were given the choice to use a modified user interface that allows users to signal their choice using the PIN pad keys instead of using the keys located on both sides of the screen.

The use of the modified user interface affects the time taken to select an operation because users no longer need to move their arms (elbow, shoulder, hand) from the PIN pad to the keys located on the side of the screen that are typically some 30 cm above the surface where the PIN pad is.

This modified interface does also away with the need for the opposite movement and can therefore improve user performance. This is the case of dialogues such as the yes/no (or cancel/confirm) dialogues to be answered through the green and red keys following an action where the user pressed a key on the side of the screen.

In fact, data analysis showed that the exclusive use of the PIN pad in operation selection resulted in lower task times concentrated in the lower part of the span of experimental values. The values for operation selection reported on Table 3 show that users took between 5 to 18 seconds to select an operation. The range of times corresponding to operation selection with exclusive use of the PIN pad is significantly lower with most of the values falling in the range of 5 to 13 seconds.

The modified user interface Confirmation (or cancellation) when requires the opposite displacement.

For the remaining data collected, and given the diversity of the impairments of the test users, it was no surprise that the minimum and maximum times listed on Table 3 show large differences.

Task	Min.	Max.	
PIN code input	Overall	5	19
	Code input	3	15
Operation selection	5	18	
Cash withdrawal selection	1	8	
Choice of slip printing	2	6	
Card withdrawal	No cash	9	15
	Cash	23	28
Bill payment	Overall	33	76
	Data input	30	66
	Confirmation	4	8

Table 3: Minimum and maximum times (in seconds) taken by motor disabled users at the ATM test bed.

However, on the whole, the maximum and minimum times are roughly of the same order of magnitude except in the case of PIN code input and payment of bills. The reasons behind this should be made evident so that the adoption of larger timeout values may be duly justified.

In the case of PIN code input one must bear in mind that it follows an action that is generally difficult to perform: insert the bank card into the slot. After this, motor impaired users must displace their arms and hands from the vicinity of the card slot to the PIN pad. Depending on the individual capacity, the time it takes a more severely impaired user to make this movement is necessarily longer than the time for a user with less movement constraints. Again, we must make allowance for the worst case.

The longer starts up delays in the case of the payment of bills have the same cause with a variation. In this operation the user needs to hold the bill in order to be able to read (and transcribe) the three sequences of digits on the bill. This requires the execution of several arm, elbow, shoulder and hand movements to manipulate the bill and therefore depends on the dexterity of the user.

4.3 Mentally Impaired Users

The test results produced in the trials with mentally impaired users are not significantly different from the values recorded with persons with other kinds of impairments as Table 4 shows.

Cash withdrawal times are very close to the values that are typical of average users as well as PIN code input.

However, one must make some allowance for the fact that these users are very much diverse in terms of capacities, experience, training and use of terminals.

This is shown on Table 4 by the fact that PIN code input minimum and maximum are roughly similar whereas total time varies from 2 seconds (almost no thinking) to 15 seconds of which 12 seconds correspond to the memory recalling followed by the decision to act.

Also, one must keep in mind the complicated nature of taking out the card and money for some mentally impaired persons who are not aware or forget the task sequence required to take possession of the two. In fact, some of these persons may take the card and leave without taking the money because the two happen in sequence and not at the same time.

Again, it is important that the software may make the user aware of this and give him/her time to recall the existence of two steps. This is why the timeout value for withdrawing the card and the money should be relatively higher for these users.

Almost the same must be said for confirmation dialogues. Once the data to make a mobile phone prepayment has been keyed in and the green key has been pressed, mentally impaired users may think that this is the end and focus their attention elsewhere and forget completely that a confirmation dialogue that must be answered to has shown up.

Task	Min.	Max.
PIN code input	Overall	2
	Code input	15
Operation selection	2	3
Card withdrawal	No cash	2
	Cash	10
Cash withdrawal selection	7	10
Choice of slip printing	1	3
Mobile phone prepayment	Overall	1
	Data input	6
	Confirmation	28
Bill payment	Overall	23
	Data input	22
	Confirmation	3

Table 4: Minimum and maximum times (in seconds) taken by mentally impaired users at the ATM test bed.

5. CONCLUDING REMARKS

A study of the times taken by disabled persons to perform banking operations on an ATM test bed has been presented. This paper has shown this determination to be essential for the success of extending banking operations by means of ATMs to persons with disabilities so that they will switch to ATMs. In fact, if timeout values are not adapted to their difficulties by giving these persons extended times to carry out banking operations, they will be discouraged to further use ATMs and will inevitably quit.

More than showing the need to extend timeout values, this study has also showed the new values that should be adopted and determined the differences in those values according to the kind of disability. The reasons for these differences were also made clear.

The question now stands on how to set the recommended timeout values for a given user depending on his/her disability.

This question has an immediate answer in the case of blind users since the ATM software detects this kind of user when key 5 of the Pin pad is pressed after the user keys in the PIN code. Given this, the values shown on Table 5 are the values to recommend when audio output operation mode is used. These are the minimum values that should be used for this kind of operation mode.

For the remaining users the answer depends on the scope of the modifications that will be introduced to the ATM software which were recommended by the wider study of which the timeout values study was only a part [UTL06]. To keep interaction uncomplicated one may devise a means similar to PIN pad key 5 that makes available to

the other kinds of disabled users the new timeout values as well as other increased accessibility facilities.

Task	Timeout (seconds)
PIN code input	25
Selection of audio mode	10
Audio menu selection	10
Card withdrawal	
No cash	20
Cash	40
Selection of amount to withdraw	20
Mobile phone prepayment	45
Bill payment	45
All confirmation dialogues	10

Table 5: Minimum timeout values recommended for operations carried out with audio output help. All times are counted from the time audio prompts finish or the user presses a key.

Task	Timeout (seconds)
PIN code input	15
Operation selection	20
Card withdrawal	
No cash	20
Cash	40
Cash withdrawal	10
Bill payment	60
Bank transfers	60
All confirmation dialogues	10
Receipt printing choice	8

Table 6: Minimum timeout values recommended for operations carried out without audio output. The above values apply to PIN pad only interaction.

Table 6 presents the recommended timeout values for operations performed by mental and motor impaired persons. For implementation simplicity, the timeouts for the two kinds of users were combined in one single table

6. ACKNOWLEDGEMENTS

The authors wish to acknowledge the involvement and participation of APD, ACAPO, ADFA, CM Cascais, CM Lisboa, CECD, Centro de Recursos para a Deficiência Visual, ME, CERCI Lisboa, Cooperativa de S. Pedro, Liga Portuguesa de Deficientes Motores, SNRIPD, UMIC, APPACDM – Marinha Grande

7. REFERENCES

- [ABNT15250] Acessibilidade em caixa auto-atendimento bancário (Accessibility in automática telling machines), Associação Brasileira de Normas Técnicas, Rio de Janeiro 2005.
- [CEN15291] Identification card system – Guidance on design for accessible card-activated devices, CEN, 2005.
- [DLW02] R. C. Davies, E. Löfgren, M. Wallergard, A. Lindén, K. Boschian, U. Minör, B. Sonesson, Three applications of virtual reality for brain injury rehabilitation of daily tasks, Proc. 4th Int. Conf. Disability, Hungary, 2002.
- [Gill98] J. Gill. Access Prohibited? Information for Designers of Public Access Terminals, RNIB Scientific Research Unit, London 1998.
- [Pereira07] L. M. Pereira, J. Brisson Lopes, A. Raposo Cordeiro e al. An example of how evaluation may change the ATM project design in Portugal, COST 219ter, to be published.
- [UTL06] L. M. Pereira, J. Brisson Lopes, A. Raposo Cordeiro et al. Condições de acesso e de utilização dos equipamentos designados por caixas automáticos (ATMs), Recomendações relativas a localização e instalação, design e construção dos equipamentos e desenvolvimento de diálogos e interação, UTL, Lisboa, 2006.

Avaliação do jogo de computador “Boccia” para adequação a uma equipa de jogadores com paralisia cerebral

Luís Carlos Bruno
ESTIG-IPBeja / LabSI / INESC-ID
Rua Afonso III, 1, 7800-050 BEJA
lbruno@estig.ipbeja.pt

João Brisson Lopes
Dep. Eng^a. Informática, IST/UTL
Av.Rovisco Pais, Lisboa
brisson@ist.utl.pt

Luís Filipe Garcia
ESTIG-IPBeja / LabSI / INESC-ID
Rua Afonso III, 1, 7800-050 BEJA
luisbgarcia@estig.ipbeja.pt

João Madeiras Pereira
Dep. Eng^a. Informática,
IST/UTL / INESC-ID
Av.Rovisco Pais, Lisboa
jap@inesc-id.pt

Sumário

Um jogo de computador 3D, que tem como finalidade permitir que jogadores portadores de paralisia cerebral possam jogar o jogo de Boccia, foi desenvolvido ao abrigo de uma tese de mestrado no Instituto Superior Técnico. Na sequência deste trabalho, foi efectuado um estudo de avaliação e adequação do referido jogo aos jogadores da equipa de Boccia do Centro de Paralisia Cerebral de Beja. Para tal, foram utilizados vários métodos, em diferentes fases, e que passaram por uma avaliação efectuada por peritos em interfaces e de jogo real, por testes de utilização dos utentes, com base em registos de observações em papel e em vídeo e, finalmente, pela aplicação de um inquérito sob a forma de entrevista aos jogadores. A experiência obtida poderá, depois, ser aplicada na adequação e avaliação de outros sistemas de software (tecnologias de apoio) a este tipo de utilizadores que apresentam necessidades especiais e que devem também ter respostas adequadas por parte destes sistemas.

Palavras-chave

Métodos de Avaliação de Interfaces, Pessoas com Necessidades Especiais, Acessibilidade, Desenho Centrado no Utilizador

1. INTRODUÇÃO

Actualmente, o desenvolvimento das tecnologias de apoio a pessoas com necessidades especiais permitem ajudar os terapeutas e os utentes na sua reabilitação, em diferentes domínios [NORTH97], [INMAN97], [CHRISTIENSEN98], como seja ao nível: motor, cognitivo, psicológico, comunicacional, entre outros. A utilização de ambientes sintéticos tridimensionais, em conjunto com o uso de dispositivos de interacção mais naturais, tem um grande potencial para promover a reabilitação das pessoas que são afectadas no seu dia-a-dia pelos seus *handicaps* [NIDRR00]. O Laboratório de Sistemas Interactivos da Escola Superior de Tecnologia e Gestão de Beja (*LabSI/ESTIG*), em conjunto com o Centro de Paralisia Cerebral de Beja (*CPCB*), têm trabalhado, nos últimos anos, em conjunto, no sentido de se criarem instrumentos que possam melhorar a qualidade de vida dos seus utentes. Nesse sentido, foram criadas algumas ferramentas de apoio a pessoas com necessidades especiais, cuja aplicação mais conhecida é o sistema “Eugénio – O Génio das Palavras”[GARCIA03], que pretende apoiar pessoas com dificuldades de comunicação, quer ao nível da escrita, quer ao nível da

fala. São também desenvolvidos regularmente estudos de avaliação de sistemas de software e hardware para adaptação aos utentes do CPCB. Uma outra área de cooperação que se está a iniciar prende-se com o desenvolvimento de ambientes virtuais que possam apoiar o processo de reabilitação e integração dos seus utentes. Neste sentido, está a ser desenvolvido um simulador virtual de uma cadeira de rodas. Deste modo, pensou-se em desenvolver um ambiente virtual que simulasse o jogo do Boccia [Boccia06], por forma a melhorar o desempenho dos jogadores da equipa de Boccia do CPCB em alguns aspectos no jogo real, em competições oficiais, e em representação do CPCB. Assim, e com base em [RIZZO98], as experiências positivas com jogos de computadores, e nomeadamente com ambientes virtuais, podem ser um factor de reforço da motivação, do desempenho e da satisfação destas pessoas, podendo estes elementos ser transferidos para situações em contexto real.

No início da análise bibliográfica sobre este tipo de simuladores identificou-se a existência de um jogo de computador que simula o jogo do Boccia e que estava a ser realizado por um aluno do mestrado do Instituto Superior

Superior Técnico, ao abrigo da sua tese (Modelação de Utilizadores Portadores de Deficiência). Assim, sabendo-se das necessidades do CPCB neste âmbito, iniciou-se um processo de avaliação e adequação do sistema, entre alguns membros do LabSI, do CPCB e do IST.

O CPCB envolveu-se neste projecto disponibilizando os seus jogadores de competição e de entretenimento que praticam o jogo Boccia, bem como alguns terapeutas (entre os quais está o treinador de Boccia) e o seu técnico de



Figura 1 – Jogo de computador Boccia

de informática.

O CPCB possui uma equipa de competição de Boccia, composta este ano por 5 jogadores, que competem a nível nacional e alguns deles já competiram a nível internacional, tendo obtido alguns títulos nesses eventos. Os principais objectivos que o treinador de Boccia pretendia atingir com um simulador de Boccia eram os seguintes: melhorar a estratégia de jogo, estimular o cálculo aritmético e funcionar como terapia ocupacional (entretenimento).

Foram disponibilizadas pelo autor do jogo três versões diferentes (protótipos funcionais do jogo). No seu conjunto, estas três versões permitem jogar em modo SinglePlayer e MultiPlayer (ambiente distribuído em rede) e permitem utilizar três diferentes dispositivos de entrada de dados (rato, teclado e manípulo (baseado na técnica do varrimento)). Possuem também um *avatar* (assistente de ajuda aos utilizadores) e um sistema de configuração de parâmetros de utilização. O jogo baseia-se num cenário sintético 3D, tal como pode ser visualizado na Figura 1 e cuja metáfora de interacção tem como base o modelo WIMP (Windows, Icons, Menus, and Pointers).



Figura 2 – Jogo em contexto real

Nos próximos capítulos descrevem-se os passos, métodos, resultados e conclusões obtidos do estudo de avaliação que foi efectuado sobre este jogo de computador de Boccia.

2. JOGO DO BOCCIA EM CONTEXTO REAL

Destinado a pessoas portadoras, ou não, de deficiência, o jogo da Boccia apareceu na Grécia antiga e consiste em aproximar bolas **azuis** e **vermelhas** de uma bola alvo (**branca**). As bolas podem ser lançadas com a mão, com o pé ou com dispositivos auxiliares (calha), para jogadores com maior dificuldade motora. Na Figura 2 fornece-se uma vista de um campo de boccia, contendo as diferentes bolas lançadas e um jogador em cadeira de rodas. Adoptado pela Cerebral Palsy-International Sport and Recreation Association [CPISRA06], o Boccia integrou pela primeira vez uma competição internacional em **1982**, nos Campeonatos do Mundo de Desporto para a Paralisia Cerebral, em Greve, Dinamarca. A modalidade faz parte do calendário para-olímpico desde **1984**.

O Boccia foi introduzido em Portugal, em Maio de **1983**, durante um curso de desporto, para paralisia cerebral organizado pela Associação Portuguesa de Paralisia Cerebral. Um ano depois, em **1984**, o jogo do Boccia integrou o I Campeonato Nacional do Desporto para Paralisia Cerebral, enquanto modalidade de demonstração. O ano de **1985** viria a ser o ano da implementação definitiva do Boccia em Portugal, através de um programa de divulgação nacional e da sua integração no II Campeonato Nacional do Desporto para Paralisia Cerebral.

3. CARACTERIZAÇÃO DOS UTILIZADORES

A equipa de Boccia do CPCB é constituída por vários jogadores pertencentes a diferentes classes de competição. Estes jogadores são todos portadores de paralisia cerebral. Esta deficiência caracteriza-se por um conjunto de desordens ao nível dos movimentos e da postura do indivíduo e ocorrem como resultado de lesões que afectam o desenvolvimento do sistema nervoso central [BAX64]. Geralmente, antes dos dois anos de vida é possível diagnosticar esta doença. Todos estes jogadores, apresentam idades compreendidas entre os 24 e 37 anos, e são utentes do CPCB, desde muito jovens. Se numa primeira fase da sua ligação à instituição, o objectivo era promover o seu crescimento e desenvolvimento físico, em idades mais avançadas o objectivo de intervenção focou-se sobretudo no seu crescimento e desenvolvimento psicossocial, bem como no seu desempenho motor [DARRAH99].

Neste sentido, o jogo do Boccia foi utilizado para desenvolver estas últimas competências, bem como uma ferramenta para ser usada na terapia ocupacional.

Inicialmente, o universo de potenciais utilizadores-alvo do jogo era composto por sete jogadores do CPCB que são portadores de paralisia cerebral. Verificou-se que um deles não poderia ser objecto de avaliação deste jogo, em virtude de realizar permanentemente movimentos involuntários, o que inviabiliza o correcto controlo dos

Jogador	Dispositivo utilizado	Modo de lançamento	Classe do Jogador	Classificação nacional 2005
LB.	Teclado	Mão	BC1	3º
SV.	Teclado	Mão	BC1	16º
AS.	Teclado	Mão	BC2	45º
FC.	Teclado c/ capacete sem grelha	Calha c/ capacete	BC3	10º
AE.	Teclado c/ capacete c/ grelha	Calha c/ capacete	BC3	S/ Ranking
FS.	Manípulo c/ varrimento	Calha c/ capacete	BC3	S/ Ranking

Quadro 1 – Lista de jogadores de Boccia do CPCB avaliados

comandos de entrada de dados, através do manípulo (interruptor). Assim, foram escolhidos os seis jogadores descritos no Quadro 1, os quais foram objecto de testes de avaliação.

As classes BC1, BC2 e BC3 referenciadas no Quadro 1 estão classificadas segundo o sistema de Classes da Associação Internacional de Desporto e Recriação para a Paralisia Cerebral (CP-ISRA) e referenciam respectivamente as classes CP1, CP2 e CP3 de atletas com paralisia cerebral, que utilizam uma cadeira de rodas durante a competição (excluindo a natação).

Os atletas CP1 (BC1) apresentam pouca amplitude funcional de movimentos e pouca força funcional em todas as suas extremidades e no tronco. Dependem de um acompanhante para ajustar ou estabilizar a cadeira de rodas ou para lhe dar a bola.

Os atletas CP2 (BC2) apresentam restrições da força funcional nos braços, pernas e tronco. Conseguem impulsionar uma cadeira de rodas e utilizam-na na competição.

Os atletas CP3 (BC3) são indivíduos com muitas dificuldades na preensão da bola e não têm força funcional para fazer um lançamento para dentro de campo. Utilizam calhas para fazer o lançamento. Nesta classe estão incluídos outro tipo de indivíduos que têm deficiências com as mesmas características funcionais.

Todos estes jogadores do CPCB locomovem-se no seu dia-a-dia, recorrendo ao uso de uma cadeira de rodas.

É interessante verificar no Quadro 1, que os jogadores que utilizam a mão para efectuarem os lançamentos da bola no jogo real, usam o teclado no jogo de computador (com recurso a teclas predefinidas para a selecção das diferentes tarefas), enquanto que os outros três jogadores que utilizam a calha no jogo real, usam diferentes dispositivos (teclado/capacete e manípulo) e diferentes métodos de selecção de opções: (i) directo com teclas pré-definidas e (ii) baseado no varrimento [COOK95].

4. METODOLOGIA DE AVALIAÇÃO

Por forma a avaliar a versão do jogo de computador Boccia foi definida uma metodologia composta por três fases. Cada uma delas é constituída por diferentes métodos e pretende obter informação útil e complementar entre si, que permita identificar a usabilidade, acessibilidade e a adequação do jogo às necessidades dos seus utentes. Como resultado da realização dessas tarefas, deve ser produzido um documento de recomendações de melhorias do sistema. Assim, uma das fases consistiu na avaliação do sistema por dois tipos de peritos: i) em interfaces de software com o utilizador e ii) em regras e treino do jogo do Boccia. Outra fase da avaliação consistiu no estudo de campo, baseado em testes de utilização por parte dos jogadores do CPCB. Estes testes permitiram obter vários elementos importantes resultantes da interacção dos jogadores com o jogo. Finalmente, e após vários meses de uso por parte dos utilizadores-alvo, foi aplicado um inquérito aos jogadores, em forma de questionário, e cujas perguntas foram intermediadas por dois terapeutas do CPCB.

4.1 Avaliação por peritos

Esta avaliação, como já foi referido anteriormente, foi desenvolvida por dois tipos de peritos. Um dos participantes neste projecto desenvolveu a avaliação das interfaces pessoa-computador, baseada em dois instrumentos: avaliação heurística dos diferentes interfaces do jogo, com base nas seguintes Heurísticas Revistas de Nielsen [NIELSEN94] e em algumas regras de acessibilidade para pessoas com necessidades especiais [IBM03]. As heurísticas de Nielsen utilizadas foram as seguintes:

- H2-1: Visibilidade do estado do sistema;
- H2-2: Correspondência entre o sistema e o mundo real;
- H2-3: Liberdade e controlo do sistema pelo utilizador;
- H2-4: Consistência e aderência a normas;
- H2-5: Prevenção de erros;
- H2-6: Reconhecimento em vez de memorização;
- H2-7: Flexibilidade e eficiência de utilização;
- H2-8: Desenho de ecrã estético e minimalista;
- H2-9: Ajudar no reconhecimento/diagnóstico/recuperação de erros;
- H2-10: Ajuda e documentação;

Pretendeu-se, com este tipo de avaliação, identificar numa fase prévia alguns problemas de interface que pudessem ser objecto de redesenho. Embora os resultados desta avaliação não tenham sido logo objecto de correcção (como deveria ser efectuado no normal ciclo de desenvolvimento das interfaces), por forma a refinar as interfaces, para a realização posterior de testes com utilizadores, serviu para complementar e validar alguns dos resultados obtidos em fases posteriores da avaliação.

No que respeita às questões de acessibilidade para pessoas com necessidades especiais, foram feitos, principalmente, testes de adequação do sistema ao nível do uso do varrimento por parte dos utilizadores. Desta forma, ten-

forma, tentou-se identificar a correcção da sequência, os tempos das opções seleccionadas, o nível de destaque e o grau de percepção das selecções das opções objecto do varrimento, de modo a adequar o sistema a alguns utilizadores que usam o manípulo para seleccionarem as opções das interfaces.

O treinador do jogo de Boccia do CPCB constituiu-se como perito de avaliação do jogo, por forma a aferir o seu grau de verosimilhança com o jogo real. Assim, e após várias sessões de utilização do jogo e de observação de jogos realizados pelos jogadores, o treinador identificou algumas questões pertinentes do jogo que devem ser refinadas. Assim, o cumprimento de certas regras do jogo e certos aspectos relacionados com a semelhança entre os lançamentos no jogo real e no virtual foram objecto de análise e recomendações por parte do treinador do CPCB.

4.2 Testes de utilização

Os testes com utilizadores finais foram realizados em várias etapas. Assim, inicialmente foram efectuadas sessões de aprendizagem e adaptação ao jogo de computador e depois foram realizadas sessões de jogos entre os vários jogadores do CPCB que integram a equipa de Boccia. Foram também realizados registos de observação de situações de jogo real, durante os três meses em que decorreram os testes com utilizadores, por forma a aferir-se a variação do desempenho dos jogadores em contexto real, na sequência da utilização do protótipo do Boccia.

Durante o mês de Novembro/2005, foram efectuadas sessões de aprendizagem e adaptação ao jogo, por parte de 6 jogadores do CPCB.

No processo de adaptação dos utentes ao protótipo, foi inicialmente explicado aos jogadores o ambiente visual do jogo e as respectivas funções de entrada de dados associadas aos botões. Foi-lhes igualmente explicado os diferentes tipos de dispositivos de interacção que podiam ser utilizados (rato, teclado, manípulo (interruptor)), bem como as diferentes técnicas de selecção de opções: método directo e varrimento. Para explicar melhor os conceitos anteriores, foram feitas demonstrações do jogo entre o terapeuta/treinador dos jogadores e o autor deste estudo. Para tal, as sessões do jogo foram projectadas numa parede da sala e os utentes puderam visionar as acções dos jogadores e os respectivos resultados visuais.

Uma tarefa que teve de ser inicialmente executada foi a identificação dos dispositivos de entrada de dados mais adequados para cada jogador. Como as avaliações psicomotoras existentes no CPCB eram insuficientes relativamente às suas competências e em relação à sua interacção com o computador, foi necessário estudar o acesso destes utilizadores aos dispositivos.

Assim, aos utilizadores que utilizam a mão para lançar no jogo real foi proposto, inicialmente, utilizarem o rato para controlar as entradas de dados do jogo de computador. Verificou-se que esses jogadores não tinham os movimentos da mão e dos dedos suficientemente precisos para controlarem correctamente o movimento e o pressionamento do botão esquerdo do rato. Verificou-se

também que utilizando o teclado, a precisão, o desempenho e o grau de satisfação desses jogadores aumentaram apreciavelmente.

No que se refere aos jogadores que utilizam a calha para jogar no jogo real, e visto que o controlo físico-motor dos movimentos das suas mãos é bastante deficiente, foi utilizado o capacete com um ponteiro para pressionar as teclas do teclado. Um destes utilizadores, pelo facto de realizar movimentos menos precisos com a cabeça, utiliza uma grelha sobre o teclado para acertar mais facilmente na tecla que pretende pressionar. É de referir que estes utilizadores já utilizavam estes dispositivos nas suas interacções com outros programas de computadores.

Finalmente, uma das utentes usou na interacção com o jogo de computador Boccia um manípulo como dispositivo de entrada de dados e a técnica de selecção por varrimento. Esta utilizadora realiza, permanentemente, bastantes movimentos involuntários, daí que nem o rato, nem o teclado podem ser controlados adequadamente pela jogadora. Uma tarefa que foi necessária realizar prende-se com a identificação dos tempos adequados de fixação de cada selecção de uma opção/botão no processo de varrimento. Para tal, através de um processo de tentativa e erro, obteve-se o valor de 5s como o tempo ideal. Valores acima deste provocam aborrecimento e desconcentração por parte da utilizadora, e valores inferiores, pelo facto de serem demasiado rápidos, não permitem geralmente que a mesma escolha a opção em tempo útil.

Foi definido um calendário para os utentes executarem as suas sessões de jogo no computador. Assim, durante os meses de Dezembro/2005 e Janeiro/2006 foi previsto serem efectuadas três sessões semanais (2ªF., 3ªF., 5ªF.) de jogos entre os vários jogadores, em que cada sessão ocupava duas horas.

O facto de se ter criado um “campeonato” durante esses dois meses, entre os seis jogadores, motivou-os a continuarem a explorar o jogo, e dessa forma foi possível obter da parte deles várias sugestões, bem como foi possível identificar as suas dificuldades e os erros ou omissões do protótipo. Com base em observação directa, os dois terapeutas que estavam com eles, ou os autores deste estudo, registavam as informações pertinentes, resultantes das acções dos utilizadores. Durante as suas sessões de jogo, os jogadores eram regularmente questionados sobre certas decisões que tomavam no jogo, dificuldades que sentiam ou até sobre a pontuação dos seus parciais.

Por outro lado, foi utilizado o registo de vídeo como mais um recurso importante para identificar a reacção dos utilizadores. Assim, o visionamento dos vídeos das várias sessões de jogos realizadas entre os jogadores, por parte dos terapeutas e do autor do estudo, permitiram identificar algumas outras questões importantes relacionadas com a interacção dos jogadores com o jogo.

Foi também realizado um outro estudo experimental com os utilizadores, que incidiu sobre o registo de observações sobre situações do jogo real, efectuado em documento próprio para o efeito. Assim, pretendia-se verificar

verificar qual era o grau de evolução dos jogadores no jogo real, ao mesmo tempo que utilizavam o jogo de computador. As situações de jogo real que se pretendiam medir foram definidas de acordo com os seguintes critérios:

- Orientação do lançamento
- Precisão do lançamento
- Bloqueios
- Tabelas

Por sua vez, cada um destes critérios era testado com base nos seguintes parâmetros de lançamento:

- Alcance: curto, médio e longo
- Direcção: esquerda, direita e centro

Assim, em Novembro de 2005 (antes do início das sessões de testes com o protótipo) foi efectuado um registo de observação dos jogadores em jogo real. Em cada um dos três meses seguintes foi efectuado um registo de observação, cuja finalidade seria verificar se o desempenho dos jogadores era ou não influenciado pela utilização do jogo de computador.

4.3 Questionários de opinião

Por forma a obter informação mais estruturada e sistematizada por parte dos utilizadores, foi desenvolvido e aplicado um questionário após a realização de todos os testes de utilização (Fevereiro de 2006). O questionário foi dividido em 4 grupos de questões:

- Informação geral sobre o jogador: idade, sistema de classificação e tipo de dispositivo de interacção, entre outros
- Grupo I: Realismo do jogo - visual, regras e interacção entre objectos
- Grupo II: Grau de facilidade de utilização do jogo de computador
- Grupo III Satisfação do utilizador

O inquérito foi desenvolvido em conjunto com o treinador por forma a haver uma adequação das questões aos jogadores. Os inquéritos foram aplicados sob a forma de entrevista a cada um dos jogadores, por parte do treinador e de uma terapeuta. Em virtude dos jogadores não saberem ler ou escrever, foi necessário intermediar o processo de recolha de respostas. Foram tidas em atenção algumas regras no processo de elaboração dos inquéritos e da realização das respectivas entrevistas [BUHLER93]:

- Evitar que os entrevistadores conduzam as respostas dos jogadores e que os levem, mesmo que involuntariamente, a seguir as suas ideias;
- Utilizar expressões e vocabulário que estejam o mais próximo possível dos jogadores;
- Evitar utilizar questões na forma negativa, mesmo que pareça ser a forma mais adequada para a percepção da pergunta por parte dos jogadores;
- Reforçar e confirmar as respostas dadas pelos jogadores.

As questões colocadas foram de escolha múltipla. Tome-se a seguir o exemplo de uma questão do questionário:

1. **Questão:** O desenho do campo (linhas, proporção das dimensões, casas de lançamento) no jogo do computador é **semelhante** ao do jogo real ?

Foi utilizada uma escala para cada resposta, baseada em três valores diferentes. A escolha desta escala foi determinada em conjunto com os terapeutas, que, com base em testes anteriores, verificaram que com mais do que três opções de resposta, os utentes do CPCB têm dificuldades em escolher uma das opções. Por outro lado, foi associado a cada questão um campo aberto que permite a introdução de alguma informação descritiva sobre a questão anterior, que permita contextualizar ou reforçar o sentido da resposta dada.

5. RESULTADOS

Esta secção pretende explicitar os principais resultados obtidos durante as várias fases de avaliação/adequação do jogo de computador.

5.1 Avaliação por peritos

No que concerne à avaliação heurística, foram identificados onze problemas nas interfaces com o utilizador, com diferentes níveis de severidade e, conseqüentemente, com diferente impacto na interacção deste tipo de utilizadores com necessidades especiais. Associada a cada problema foi também incluída uma sugestão de resolução do mesmo. Tome-se o seguinte exemplo:

Problema: Na interface do jogo em execução, quando um jogador está a jogar, não é possível identificar o nº de bolas que o seu adversário ainda tem disponíveis para jogar. Somente é possível identificar as restantes bolas do jogador actual (bolas azuis ou vermelhas).

Heurísticas violadas: H2-2: Correspondência entre o sistema e o mundo real, H2-6 – Reconhecimento em vez de memorização.

Sugestão de Resolução: Mostrar de forma permanente na interface com o utilizador o número de bolas que ainda faltam ser jogadas, por qualquer um dos dois jogadores (bolas azuis e vermelhas).

No que respeita à avaliação efectuada por parte do treinador, foram efectuadas onze recomendações de alteração de diferentes aspectos do sistema, por forma a seguirem as regras oficiais, como sejam as sequências dos parciais, os tempos de jogo e a escolha do jogador a lançar, entre outros. Assim, tome-se o seguinte exemplo de recomendação: após o lançamento de uma bola, a linha que indica a sua direcção e alcance deve ser removida. Caso contrário, tal como acontece nesta versão do jogo, o lançamento seguinte tem por omissão a mesma direcção e alcance do anterior. Desta forma não são exploradas, como se pretendia que o jogo possibilitasse, todas as competências de lançamento por parte dos jogadores.

Foram também feitas recomendações de adaptação do jogo de computador às regras oficiais, por forma a dar-lhe um carácter mais verosímil para os jogadores. O não cumprimento de algumas regras do jogo provocou nos utilizadores algumas reacções de desagrado, visto ter algumas contradições com as regras do jogo real.

5.2 Testes de utilização

Como foi referido anteriormente, os jogadores realizaram um campeonato, no qual jogaram quase todos uns contra os outros. Na sequência desses jogos, cerca de cem jogos parciais foram registados alguns resultados interessantes:

- Os jogadores que estão inscritos na competição oficial do jogo real são os que realizaram mais jogos e os que obtiveram mais vitórias;
- O jogador com melhor ranking no ano 2005 do jogo real foi o que obteve mais vitórias;
- O FC, embora jogue com capacete, obteve mais vitórias sobre os jogadores que jogam com o teclado, do que estes em relação àquele.

Da análise visual dos jogos realizados, verificou-se que os jogadores abordam as soluções das várias situações de jogo no protótipo, de forma muito semelhante às correspondentes no jogo real. Situações como o lançamento da bola inicial ou as tabelas para retirar bolas da frente são exemplos disso.

No que concerne ao registo de observações sobre situações do jogo real, efectuado em documento próprio, com base nos critérios de lançamento: orientação do lançamento, precisão do lançamento, bloqueios e tabelas, foram registadas as sessões dos meses de Novembro/2005, Dezembro/2005 e Janeiro/2006, dos quatro jogadores que participam em competições oficiais.

Cada lançamento é pontuado pelo treinador numa escala de 0 a 2:

- 0 - corresponde a um mau lançamento
- 1 - corresponde a um lançamento razoável
- 2 - corresponde a um bom lançamento

Após os jogadores terem treinado com o jogo de computador durante esses três meses, verificou-se que todos os jogadores melhoraram em média as pontuações dos seus lançamentos no jogo real, entre cada dois meses consecutivos e consequentemente entre o mês inicial e o final. Registou-se uma melhoria em média de 0,37 (na escala de 0 a 2) na média dos lançamentos dos quatro jogadores. É claro que os resultados apresentados podem não justificar a relação de causa/efeito entre o uso do jogo de computador e estes resultados, durante esses três meses. Para que tal acontecesse teríamos que ter um outro grupo de jogadores de controlo, que não utilizasse o jogo de computador, mas que treinassem o jogo real nas mesmas condições. Assim, poderíamos comparar os desempenhos dos jogadores dos dois grupos e verificar a influência do jogo de computador. Outra variável, que influencia a curva de aprendizagem dos utilizadores e permite evidenciar ou não a sua consolidação, prende-se com o tempo de utilização do sistema e o respectivo registo de observações. Durante um período mais longo (por exemplo de doze meses) será possível obter dados mais fiáveis e esclarecedores sobre o uso do sistema.

5.3 Questionários de opinião

O inquérito realizado foi dividido em vários grupos de questões. Os resultados produzidos expressam a frequência de respostas dos utilizadores, para cada um dos três

dos três valores da escala (um a dois valores). Por exemplo, para a questão 1, do grupo I, 5 utilizadores pontuaram a questão com o valor máximo de aceitação, e 1 utilizador pontuou a questão com o valor médio de aceitação.

Da análise das pontuações dos jogadores a cada questão, devem-se tirar ilações sobre a hipótese ou não de redesenho do sistema para essas questões:

- Se a maioria das pontuações for no valor 3, **não deverá** haver necessidade de efectuar revisão do sistema para essa questão;
- Se a maioria das pontuações for no valor 2, **poderá** haver necessidade de efectuar melhoramento do sistema para essa questão;
- Se a maioria das pontuações for no valor 1, **deverá** haver necessidade de efectuar revisão do sistema, para essa questão.

A seguir são mostradas as questões colocadas aos jogadores e os resultados obtidos.

Grupo I (Realismo visual, regras e interacção entre objectos)

Q1: O campo é semelhante ao do jogo real?

Q2: A colisão das bolas é realista?

Q3: O jogador pode efectuar os mesmos movimentos que no jogo real?

Q4: O campo de visão sobre o campo é semelhante ao do jogo real?

Q5: A sequência dos jogos parciais é semelhante à do jogo real?

No Quadro 2 podem-se verificar os resultados obtidos nas questões do grupo I.

Pontuação	1	2	3
Questão 1	0	1	5
Questão 2	6	0	0
Questão 3	0	6	0
Questão 4	0	6	0
Questão 5	6	0	0

Quadro 2 – Resultados do grupo I de questões

Grupo II (Facilidade de utilização do jogo de computador)

Q7: Os ícones dos botões identificam a função associada?

Q8: O posicionamento dos botões de lançamento das bolas é correcto?

Q9a: Existe facilidade de selecção dos botões com rato?

Q9b: Existe facilidade de selecção dos botões com teclado ?

Q9c: Existe facilidade de selecção dos botões com teclado com capacete/ponteiro?

Q9d: Existe facilidade de selecção dos botões com switch e com varrimento?

Q10: Percebe-se a sequência do varrimento?

Q11: Percebe-se que uma opção está seleccionada com varrimento?

No Quadro 3 podem-se verificar os resultados obtidos nas questões do grupo II.

Pontuação	1	2	3
Questão 7	0	4	2
Questão 8	0	2	4
Questão 9a	0	0	0
Questão 9b	0	0	3
Questão 9c	0	0	1
Questão 9d	0	1	1
Questão 10	0	0	1
Questão 11	0	0	1

Quadro 3 – Resultados do segundo grupo de questões

Grupo III (Satisfação do utilizador)

Q13: Qual é a utilidade do jogo para treinar Boccia?

Q14: Qual é a utilidade do jogo para lazer?

Q15: Gosta de jogar?

Q16: Pretende continuar a jogar após implementação de melhoramentos?

No Quadro 4 podem-se verificar os resultados obtidos nas questões do grupo II.

Pontuação	1	2	3
Questão 13	0	3	3
Questão 14	0	0	6
Questão 15	0	0	6
Questão 16	0	0	6

Quadro 4 – Resultados do terceiro grupo de questões

Verifica-se pelos resultados obtidos que as questões Q2 e Q5 **deverão** ser objecto de redesenho do sistema e as questões Q7 e Q13 **poderão** ser objecto de análise para posteriores melhoramentos.

6. CONCLUSÃO E TRABALHO FUTURO

Procedeu-se à avaliação e adequação de um sistema de software, por forma a que os seus utilizadores, que neste caso possuem necessidades especiais, possam interagir com ele de uma forma mais natural e eficaz. Para tal fo-

ram envolvidos jogadores, treinador e terapeutas neste processo, o que permitiu, de uma forma mais integrada, receber o contributo de todos no sentido da melhoria deste jogo de computador Boccia.

Verificou-se que a aprendizagem do ambiente de utilização do jogo e a percepção do seu ambiente visual, por parte dos utentes do CPCB, decorreu de forma bastante satisfatória. Os jogadores mostraram sempre um grande entusiasmo no uso do jogo de computador, o que provavelmente lhes permitiu transferir essa motivação para os treinos do jogo em contexto real. É de referir, embora possa não haver uma relação de causa/efeito, que este ano a equipa de Boccia do CPCB, e contrariamente aos últimos 4 anos, passou à fase final do campeonato nacional de Boccia.

As correcções/recomendações a efectuar no jogo basearam-se nos resultados de várias técnicas de avaliação: avaliação por peritos (interfaces de software e técnica do jogo real); testes com utilizadores e respectiva observação e registo de execução de tarefas; e a aplicação de inquéritos sob a forma de entrevistas. Verifica-se que a conjugação no uso destas técnicas permite identificar de forma mais integrada os elementos críticos na interacção do sistema com este tipo de utilizadores. Algumas vezes, a conjugação destas técnicas permite validar e reforçar alguns resultados obtidos.

Como se pode verificar nas respostas às questões “**Q15:** Gosta de jogar este jogo de computador?” e “**Q16:** Pretende continuar a jogar após implementação de melhoramentos?”, os utilizadores aderiram ao uso do jogo e gostariam de continuar a utilizá-lo após os melhoramentos.

Verifica-se, com este estudo, a importância de centrar o desenvolvimento destes projectos nos utilizadores-alvo, sendo para tal necessário acolher as suas contribuições e necessidades, desde o seu início, passando por todas as outras fases, entre as quais os momentos de avaliação.

Pretende-se que proximamente as recomendações efectuadas possam ser objecto de redesenho e respectiva implementação neste jogo de computador, para posteriormente se procederem a novos testes que permitam chegar a um estado de maturidade adequado na interacção do sistema com os utilizadores. Chegados a este estado, podem-se então estudar alguns critérios relacionados com a evolução do comportamento dos jogadores em contexto do jogo real, após o uso do jogo de computador, como sejam: tempo de reacção, tempo de decisão, nível de desinibição e grau de cooperação entre os utilizadores.

7. AGRADECIMENTOS

Agradece-se à direcção do CPCB a sua disponibilidade institucional a qual permitiu a realização deste trabalho, e, em especial, ao terapeuta/treinador de Boccia João Garcia e ao técnico de informática António Mendes, que foram sempre solícitos nas respostas aos pedidos efectuados.

8. REFERÊNCIAS

- [BAX64] BAX, M. Terminology and classification of cerebral palsy. *Developmental Medicine and Child Neurology*, 6, 295-297, 1964.
- [BOCCIA06] International Boccia Committee Website Homepage
<http://www.bocciainternational.com/>
- [BUHLER93] Buhler, C. and Schmidt, M. User involvement in evaluation and assessment of assistive technology. ECART 2. Proceedings of the European conference on the advancement of rehabilitation technology, 26 - 28 May 1993, The Swedish Handicap Institute Stockholm, Sweden Section 30.1 3pp. 1993
- [CHRISTIANSEN 98] CHRISTIANSEN, C.; ABREU, B.; OTTENBACHER, K.; HUFFMAN, K.; MASEL, B.; CULPEPER, R. "Task Performance in Virtual Environments used for Cognitive Rehabilitation after Traumatic Brain Injury"; *Archives Phys. Med. Rehabilitation*, V. 79, pp. 888-892, 1998
- [COOK95] COOK, Albert M. e Susan M. Hussey. *Assistive Technologies: Principles and Practice*. Mosby, Missouri. 1995
- [CPISRA06] Cerebral Palsy-International Sport and Recreation Association Website Homepage
<http://www.bocciainternational.com/>
- [DARRAH99] DARRAH., WESSEL., NEARINGBURG, P., et al. Evaluation of a community fitness program for adolescents and children with cerebral palsy. *Pediatric Physical Therapy*, 11: 18-23, 1999
- [GARCIA03] Garcia, Luís. *Concepção, Implementação e Teste de um Sistema de Apoio à Comunicação Aumentativa e Alternativa para o Português Europeu*. Tese de Mestrado. Instituto Superior Técnico. 2003
- [IBM03] IBM software accessibility checklist Webpage
<http://www-306.ibm.com/able/guidelines/software/accesssoftware.html>
- [INMAN97] INMAN, D.; LOGE, K.; LEAVENS, J. "VR Education and Rehabilitation", *Communications of the ACM*, 40(8), pp 53-58, 1997
- [NIDRR00] National Institute on Disability and Rehabilitation Research; 2000
<http://gcs.ed.gov/fedreg/announcement.html>.
- [NIELSEN94] Nielsen, J. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY. 1994
- [NORTH97] NORTH, M.; NORTH, S.; COBLE, J. "Virtual Reality Therapy: An Effective Treatment for Psychological Disorders", *Virtual Reality in Neuro-Psycho-Physiology*; Ed.G.Riva, IOS Press, Holland. 1997
- [RIZZO98] RIZZO, A. A., BUCKWALTER, J. G., NEUMANN, U., et al. Basic issues in the application of virtual reality for the assessment and rehabilitation of cognitive impairments and functional disabilities. *CyberPsychology and Behavior*, 1: 59-78, 1998

O Sítio do Pico: um espaço de aprendizagem para crianças com paralisia cerebral

Anabela Cardoso da Silva Elisabete Sousa Cunha
Pedro Henriques Sandra Lopes
Departamento de Informática
Universidade do Minho, Braga

{prh,sandralopes,anabelasilva,elisabetecunha}@di.uminho.pt

Resumo

O presente artigo apresenta o Sítio do PICO, um lugar de aprendizagem para crianças com paralisia cerebral. Estas crianças manifestam sérias dificuldades em comunicar e por isso é-lhes ministrado um processo educativo especial, adequado, baseado num Sistema Alternativo e Aumentativo de Comunicação (SAAC) no qual, entre outras coisas, se ensinam alfabetos visuais (pictóricos) que facilitam a dita comunicação. Nesse contexto surge este sistema de apoio ao Ensino/Aprendizagem de Alfabetos Alternativos, que visa oferecer um ambiente educativo adaptado às características do aprendiz (afectado por paralisia cerebral). Para isso, o Sítio do PICO usa comunicação visual e apoia-se na técnica de varrimento para selecção de opções. Além destas características genéricas, a forma como o Sítio do PICO foi implementado, possibilita a sua adaptação a cada caso (aprendiz, ou situação de aprendizagem) pois permite que os profissionais do ensino especial interactuem com a aplicação de modo a enriquecê-la e personalizá-la. Todo este projecto foi desenvolvido com a cooperação dos especialistas em tecnologia e educação especial.

Palavras-Chave

Software Educativo, Programação Web, Sistemas de Informação

1. INTRODUÇÃO

A paralisia cerebral é uma condição caracterizada por um mau controle muscular, espasticidade, paralisia e outras deficiências neurológicas decorrentes de uma lesão cerebral que ocorre durante a gestação, durante o nascimento, após o nascimento ou antes dos 5 anos de idade. Algumas crianças têm perturbações ligeiras, quase imperceptíveis, que as tornam desajeitadas a andar, falar ou a usar as mãos. Outras são gravemente afectadas com incapacidade motora grave, impossibilidade de andar e falar, sendo dependentes nas actividades da vida diária. Entre este dois extremos existem os casos mais variados. De acordo com a localização das lesões e áreas do cérebro afectadas, as manifestações podem ser diferentes. Para ajudar estas crianças, com dificuldades profundas a escrever/ler e falar, a aumentar a sua autonomia, é-lhes ensinado, logo que possível, um Sistema Alternativo e Aumentativo de Comunicação (SAAC), nomeadamente um alfabeto visual constituído por símbolos pictóricos [Sequeira93].

Seguindo as regras de ensino adoptadas pelos terapeutas, O Sítio do PICO — uma aplicação informática baseada em tecnologia Web (essencialmente **Flash** (para toda a camada interactiva e lógica do sistema), e **MySQL** (para gestão da base de dados), com recurso ao **Php** para ligação entre os anteriores) — tem por finalidade apoiar os educadores no ensino do referido

SAAC [Amorim05, Pires05, Silva06]. O Sítio do PICO funcionará, assim, como mediador na realização de actividades educativas que permitem a formação destas crianças em áreas distintas do conhecimento. Além de oferecer diversas actividades que ajudam o aprendiz a conhecer e empregar os símbolos, o próprio sistema usa, sempre que possível, o mesmo alfabeto pictórico na sua interface de comunicação com o utilizador.

A tecnologia (**Flash, MySQL, Php**) utilizada neste projecto tem a vantagem de ser de baixo custo, o que veio enriquecer as instituições educativas, como por exemplo a **APPC**, Associação Portuguesa de Paralisia Cerebral, que se deparam com bastantes obstáculos quando pretendem adquirir *software* educativo (geralmente muito caro e pouco versátil). O facto de cada um dos seus aprendizes ser um caso particular faz com que haja necessidade de comprar um vasto leque de *software* que satisfaça todas as suas carências, o que torna ainda mais pesada a referida dificuldade financeira e demonstra a relevância de se encontrar uma solução barata.

O programa desenvolvido distingue-se dos habituais — por exemplo do **Boardmaker** comercializado em Portugal pela **Anditec** (ver www.anditec.pt ou www.mayer-johnson.com), no qual este se inspirou — porque não é um sistema estático, permitindo ao educador

adaptar, otimizar e acrescentar novas situações de aprendizagem.

Assim, o **Sítio do PICO**, que será brevemente introduzido na secção 2, organiza-se em dois subsistemas: o Sistema de Aprendizagem (**SA**) pretendido, descrito na secção 3; e o Sistema de Gestão da Aprendizagem (**SGA**) referido na secção 4.

2. O SÍTIO DO PICO: APRESENTAÇÃO DO SISTEMA

Quando se desenvolve um sistema em que os principais utilizadores são aprendizes com paralisia cerebral, tem de haver uma preocupação acrescida com o design gráfico das suas páginas [Ribeiro96]. Assim, na concepção do **Sítio do PICO**, apesar de se pretenderem páginas atractivas (apelativas, leves, não-monótonas), foram utilizadas apenas três cores, visto que demasiadas cores desviam a atenção dos aprendizes. As imagens têm dimensões maiores que numa interface vulgar, pois alguns aprendizes apresentam problemas de visão. Do ponto de vista gráfico, estas foram os principais preocupações.

As crianças com paralisia cerebral manifestam perturbações motoras graves o que torna difícil a sua interacção com o computador, dado que não têm capacidade para utilizar o teclado e o rato normal. Assim, para facilitar a interacção aprendiz-computador, o nosso sistema utiliza uma técnica de varrimento (que nos foi pedida inicialmente) que consiste em destacar visualmente um a um os objectos gráficos que podem ser seleccionados, de acordo com uma determinada sequência a uma certa cadência e ciclicamente. No nosso caso utilizámos uma moldura vermelha como forma de destacar. Quando a moldura estiver sobre o objecto que corresponde à opção desejada o educando apenas dá um clique num rato especial, criado para o efeito, podendo o cursor estar posicionado em qualquer parte do ecrã. Desta forma as duas acções, seleccionar e clicar na opção pretendida, são substituídas por um único clique.

Ao criar os objectos gráficos da camada de interface achámos igualmente importante utilizar símbolos pictográficos do alfabeto **SPC** (Símbolos Pictográficos para a Comunicação), sempre que possível, para haver coerência com o que se pretende ensinar (ver figura 1). Um trabalho de grande envergadura, nesta área de desenvolvimento *software para a Educação Especial*, que merece aqui referência embora não possa ser comparado directamente com o **Sítio do PICO**, é o projecto **Fressa 2006** (ver www.xtec.es/~jlagares/index.htm) levado a cabo em Girona pela **Toti PM** (ver www.xtec.es/~jlagares/index.htm), da responsabilidade de Jordi Lagares.

3. O SÍTIO DO PICO: FUNCIONALIDADES DO SISTEMA

Nesta secção vamos descrever as funcionalidades desenvolvidas e que são disponibilizadas pela aplicação, debaixo de três grandes grupos: Jogos, Histórias e Estudar.

3.1. Jogos

No pacote jogos temos:

1. jogos de Matemática;
2. jogos de Frases.

Nos jogos de Matemática temos:

1. jogos de números, nestes utilizamos imagens às quais está associado um número; os educandos têm que saber qual o número correcto correspondente à imagem (ver figura 2);
2. jogos de somar e de subtrair, neste caso o educando deverá seleccionar o número correspondente ao resultado da adição ou subtracção das quantidades representadas em cada uma das imagens (ver figura 3).

Nos jogos de frases, conforme mostra a Figura 4, é apresentada uma cena, um enigma e uma lista de hipóteses. A cena representa informação útil para que o educando chegue à solução com sucesso. O enigma é a frase incompleta (onde existem espaços em branco em vez dos símbolos), ou seja, é a questão que é colocada ao educando. A solução é a resposta ao enigma. A lista de hipóteses contém o conjunto de símbolos com o qual o educando obtém a solução completando os espaços em branco do enigma. Para tornar o jogo mais rico, do ponto de vista da aprendizagem, a lista de hipóteses contém, além dos símbolos que constituem a solução, outros símbolos do alfabeto em uso.

3.2. Histórias

Nas histórias temos:

1. Construtor de histórias.
2. Contador de histórias.

No construtor de histórias (ver figura 5) temos as imagens que formam a história, dispostas por uma ordem aleatória.



Figura 1. Interface do Sítio do PICO

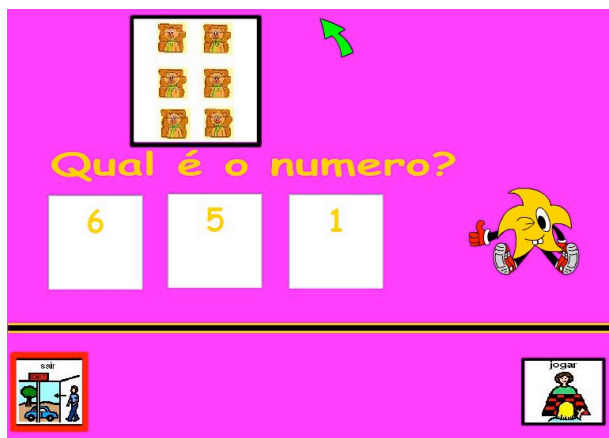


Figura 2. Jogo de Matemática: aprender os números

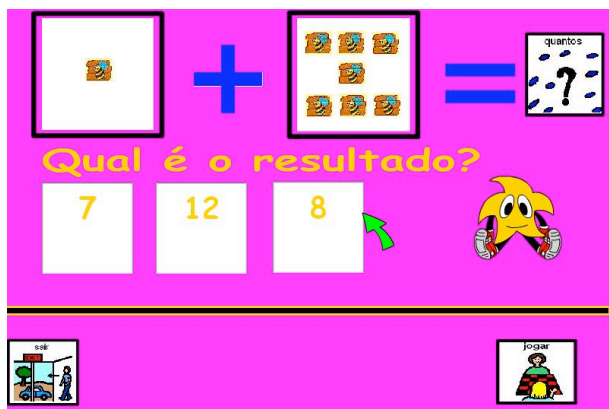


Figura 3. Jogos de Matemática: aprender a somar



Figura 4. Jogos de frases

O educando terá que analisar as imagens e dispo-las por ordem correcta.

Adicionalmente é possível despoletar a narração da história como técnica de reforço à aprendizagem.

No contador de histórias temos a possibilidade de ouvir uma voz off a ler a história que está a ser apresentada através de uma sequência de imagens.



Figura 5. Construtor de histórias

3.3. Estudar

No pacote estudar vamos ter:

1. Estudo do alfabeto;
2. Estudo dos animais.

A primeira opção tem o objectivo de levar as crianças a aprender as letras do alfabeto tradicional associando-as ao nome de um objecto e ao símbolo pictográfico correspondente (reforço da aprendizagem).

A segunda opção é também um jogo associativo destinado a ensinar os aprendizes a identificar os animais. De novo para cada um será mostrado o respectivo nome e os símbolos pictográficos correspondentes.

4. PÁGINA DA ADMINISTRAÇÃO

Neste sistema toda a informação, necessária à construção de cada actividade, está armazenada numa base de dados. O educador pode aceder à Base de Dados, através do SGA, alterando, removendo e inserindo informação para aumentar as actividades associadas a um educando ou tornar o sistema utilizável com outros educandos. A figura 6 mostra o ecrã de entrada no Sistema de Administração; vê-se nesse ecrã a lista das entidades que podem ser manuseadas pelo educador. A figura 7 ilustra as operações disponíveis sobre cada conjunto de tabelas da base de dados correspondentes a cada uma das 11 entidades do sistema de informação (neste caso foi seleccionada a entidade Cenas). Sendo este subsistema apenas usado por educadores e responsáveis pela manutenção do sistema, a interacção processa-se nos moldes tradicionais, como se percebe nos dois ecrãs acima. Contudo não deixamos de o

referir aqui para ilustrar as diferentes opções de interacção tomadas e, também, por considerarmos que esta componente é uma das características que mais distingue o **Sítio do PICO** dos sistemas concorrentes (comerciais) que até à data encontramos.

5. CONCLUSÃO

O objectivo principal a ter em conta quando se inicia uma criança num Sistema Aumentativo de Comunicação é torná-la capaz de o utilizar de uma forma espontânea e eficaz. Ou seja, deve-se promover a interacção comunicativa de forma a permitir que a criança inicie espontaneamente a conversação e responda concretamente às questões que lhe são postas [Vieira96].

Neste contexto e com base no método de trabalho praticado pelas terapeutas da **APPC**, o **Sítio do PICO**, apresentado ao longo deste artigo, foi concebido e implementado com o intuito de ser suficientemente simples de usar, atractivo e, simultaneamente, versátil podendo ser configurado para se adaptar às necessidades de cada criança. O **Sítio do PICO** permite, através de actividades lúdicas—como a *composição directa de frases simples que descrevem uma cena*, ou a *organização de imagens soltas para formar uma pequena história*—encadeadas de forma programada, ou organizadas na hora, ensinar a perceber e usar os símbolos do alfabeto visual **SPC** ou mesmo a ler o alfabeto textual clássico. Possibilita, através de jogos matemáticos, o ensino dos algarismos, da noção de contagem e de algumas operações algébricas rudimentares. Também faculta a aprendizagem da fauna e da flora e, ainda, oferece a oportunidade de ouvir contar pequenas histórias.

Para facilitar uma interacção acessível a crianças com fortes deficiências motoras, a interface usa uma manípulo especial (semelhante a um rato com 1 só botão) e recorre ao método de varrimento para selecção das opções. Esta solução de *interacção por varrimento* foi um dos principais requisitos impostos desde o início pelos utilizadores finais do sistema, mas pode obviamente ser posta em causa e reconsiderada em desenvolvimentos futuros do **Sítio do PICO**, caso se consiga formar a equipa multi-disciplinar alargada, de investigação aplicada, que um projecto desta envergadura exige e merece e que andamos a tentar constituir.

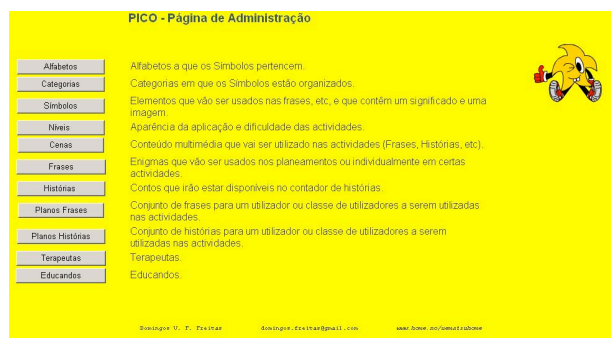


Figura 6. Página de Entrada no subsistema de Administração

É ainda de salientar o facto de o **Sítio do PICO** poder ser disponibilizado a muito baixo custo, graças à solução tecnológica adoptada baseada em ferramentas de desenvolvimento para a Web como **Flash**, **MySQL** e **Php**.

A versão actual do **Sítio do PICO**, cumprindo todos os requisitos básicos que nos solicitaram e nós próprios nos impusemos, foi entregue às terapeutas da **APPC** de Guimarães no início do verão de modo a que possa ser testada com os alunos destinatários a partir do próximo ano lectivo. Dessa utilização, em ambiente real de sala de aula, resultarão de certeza diversas recomendações que esperamos poder incorporar no sistema logo de seguida.

6. AGRADECIMENTOS

Este projecto nasceu por iniciativa da Dr.^a Maria da Cruz, directora da secção de Guimarães da **APPC**, contando sempre com o apoio das terapeutas Branca Ornelas e Glória Vasconcelos. A sua implementação informática, na versão inicial, surgiu devido ao trabalho de uma equipa de alunos do Departamento de Informática da UM que, no âmbito do projecto final (Opção III) da licenciatura em Engenharia de Sistemas e Informática, concebeu e desenvolveu o primeiro protótipo do sistema: Domingos Freitas, Helena Lopes e Lília Ferreira. Agradecemos também à Raquel Pires e Marlene Amorim que desenvolveram uma segunda versão no âmbito do projecto de estágio da licenciatura em Matemática e Ciências de Computação.

7. REFERÊNCIAS E BIBLIOGRAFIA

[Amorim05] AMORIM, M. Desenvolvimento de Software Educativo para Crianças com Paralisia Cerebral: Implementação do Varrimento. Relatório de Estágio, LMCC, Universidade do Minho, Braga. (Out. 2005).

[Pires06] PIRES, R. Desenvolvimento de Software Educativo para Crianças com Paralisia Cerebral: Módulo de Administração. Relatório de Estágio, LMCC, Universidade do Minho, Braga. (Jan. 2006).

[Ribeiro96] RIBEIRO, M. et al. Comunicação na Deficiência Mental Profunda. Braga: Edições APPCDM. (1996).

[Sequeira93] SEQUEIRA, F. Linguagem e Comunicação. Comunicare. Revista de Comunicação n.º 1. (1993).

[Silva06] SILVA, A. Desenvolvimento de Software Educativo para Crianças com Paralisia Cerebral: Concepção e desenvolvimento de novos jogos didácticos. Relatório de Estágio, LMCC, Universidade do Minho, Braga. (Jul. 2006).

[Vieira96] VIEIRA, F. D. e PEREIRA, M. C. A educação das pessoas com deficiência mental. Lisboa: Fundação Calouste Gulbenkian, Serviço de Educação. (1996).

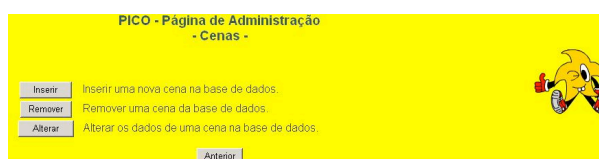


Figura 7. Página de manutenção da entidade Cenas

Concepção de um Assistente de Software para o Desenvolvimento de Teclados para o Sistema Eugénio

Alexandra Moedas Luís Garcia Luís Bruno
 LabSI/ESTIG-IPBeja
 Rua Afonso III, 1, 7800-050 BEJA
 amoedas@sapo.pt
 luisbgarcia@estig.ipbeja.pt
 lbruno@estig.ipbeja.pt

Sumário

O Eugénio é um sistema de apoio a pessoas com dificuldades de comunicação escrita ou falada. Devido ao enorme espectro de dificuldades dos seus utilizadores o sistema disponibiliza uma grande variedade de possibilidades de configuração. A configuração destas opções por parte dos técnicos de reabilitação e de professores do ensino especial, normalmente com poucos conhecimentos na área das tecnologias, torna-se por isso uma tarefa extremamente complicada e muitas vezes destinada ao insucesso. Neste artigo apresenta-se o desenvolvimento de um assistente que pretende reduzir a complexidade na realização de uma tarefa importante neste sistema; a criação de novos teclados. A experiência obtida poderá depois ser aplicada noutras tarefas de configuração do Eugénio ou noutras tecnologias de apoio.

Palavras-chave

Assistentes de Software, Desenho Centrado no Utilizador, Análise de Tarefas, Cenários de Utilização, Prototipagem, Avaliação da Usabilidade, Pessoas com Necessidades Especiais.

1. INTRODUÇÃO

Os avanços tecnológicos têm vindo a permitir o desenvolvimento de diversos sistemas de apoio a pessoas com dificuldades sensoriais, cognitivas, de comunicação e motoras. A Engenharia de Reabilitação é a área do conhecimento que se dedica ao estudo e desenvolvimento deste tipo de sistemas. A legislação Norte-Americana define esta área como *a aplicação sistemática de tecnologias, metodologias da engenharia, ou princípios científicos à satisfação das necessidades e remoção das barreiras com que se confrontam as pessoas com*

handicaps em áreas onde se inclui a educação, reabilitação, emprego, transporte, autonomia e lazer [PublicLaw86]. O sistema “Eugénio – O Génio das Palavras” foi desenvolvido pelo Laboratório de Sistemas Interactivos da Escola Superior de Tecnologia e Gestão de Beja (*LabSI/ESTIG*), Instituto de Engenharia e Sistemas de Computadores/Instituto Superior Técnico (*INESC/IST*) e o Centro de Paralisia Cerebral de Beja (*CPCB*) e pretende apoiar pessoas com dificuldades de comunicação, quer ao nível da escrita, quer ao nível da fala (Figura 1).

Para pessoas incapacitadas de utilizarem um teclado de computador este sistema dispõe de um teclado de ecrã. Este componente apresenta uma matriz contendo os vários caracteres disponíveis para a composição de mensagens. A selecção destes elementos pode ser efectuada através de métodos de acesso directo que utilizam por exemplo um dispositivo de ponteiro, ou métodos de acesso indirecto que recorrem apenas a um ou dois interruptores. Como as taxas de comunicação permitidas por estas formas alternativas são geralmente menores que as proporcionadas através dos métodos naturais o Eugénio dispõe de duas técnicas que possibilitam a aceleração da escrita, a predição de palavras e a expansão de abreviaturas. Para o reforço da interacção com o utilizador foi incorporado no sistema um agente de interface e um sintetizador de fala. Este

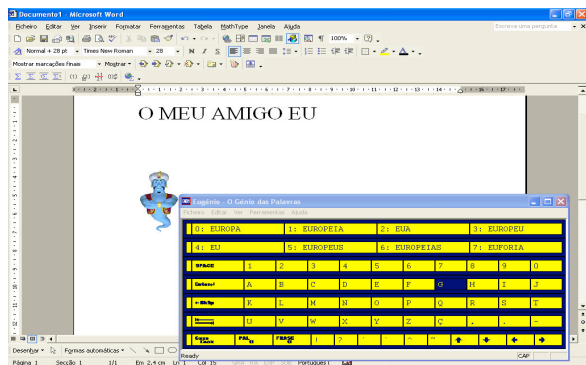


Figura 1 – O sistema “Eugénio – O Génio das Palavras”

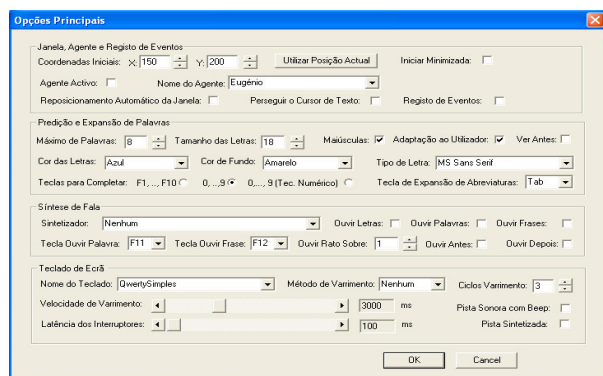


Figura 2 – Caixa de diálogo para configuração do Eugénio

último elemento também permite a transformação de uma mensagem escrita em fala e o seu envio a um interlocutor. Os vários componentes da interface do sistema podem ser adaptados às necessidades particulares de cada pessoa [Garcia03].

2. O PROBLEMA

Devido ao enorme espectro de dificuldades dos seus utilizadores, este sistema disponibiliza aos técnicos uma grande variedade de possibilidades de configuração. Exemplos de aspectos a configurar no Eugénio são o método de acesso (teclado físico, dispositivo de ponteiro ou varrimento), a disposição de teclas do teclado, o número de palavras propostas pelo sistema ou o ecoar na forma de fala cada letra ou palavra escrita pelo utilizador. A configuração destas opções por parte dos técnicos de reabilitação e professores do ensino especial, normalmente com poucos conhecimentos na área das tecnologias, torna-se por isso uma tarefa extremamente complicada e muitas vezes destinada ao insucesso. Num estudo foram analisados diversos casos de utilização deste tipo de tecnologia e verificou-se que muitas das soluções que requeriam uma configuração inicial relativamente complicada encontravam-se paradas sem qualquer tipo de utilização [Dawe06]. Pensamos que

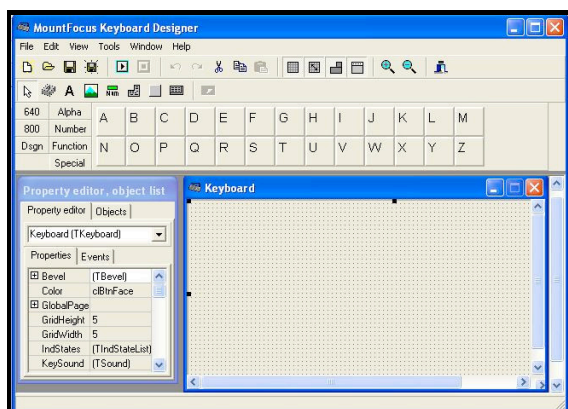


Figura 3 – Editor de teclados do sistema MountFocus

estas dificuldades podem ser ultrapassadas através do recurso a interfaces de configuração não tradicionais, que forneçam mais apoio ao utilizador do que as simples caixas de diálogo com as diversas opções disponíveis (Figura 2) ou editores que permitem por exemplo a criação de novas disposições de teclado [MountFocus03] (Figura 3). Para enfrentar as dificuldades acima referidas o LabSI da ESTIG encontra-se a estudar a utilização de assistentes de software para a realização de tarefas de configuração no contexto das tecnologias de apoio. Um assistente é uma ferramenta que conduz o utilizador passo a passo na realização de uma tarefa, recorrendo às respostas que este vai fornecendo às suas questões [Dix03]. Neste momento estão a ser desenvolvidos dois assistentes de software, um para configurar as várias opções disponíveis no sistema, outro para o desenvolvimento de novos teclados. Neste artigo os autores pretendem partilhar a sua experiência na aplicação de técnicas de interação pessoa-computador no desenvolvimento, ainda em curso, deste último assistente assim como avaliar a sua eficácia na configuração destas tecnologias para o apoio a pessoas com necessidades especiais.

3. CONCEPÇÃO DO SISTEMA

Para o desenvolvimento de um assistente de configuração realmente eficaz foi extremamente importante centrar o desenho do sistema no utilizador. Nesta abordagem foram utilizadas diversas técnicas de desenho de sistemas interactivos como o recurso a personagens representativas dos utilizadores, o desenvolvimento de cenários de utilização e a prototipagem. Podemos enumerar as principais etapas de desenvolvimento deste projecto:

- Estudo sobre sistemas semelhantes
- Caracterização dos utilizadores
- Desenvolvimento de cenários de utilização
- Desenvolvimento e avaliação de protótipos de baixa fidelidade
- Desenvolvimento e avaliação do protótipo de alta fidelidade

A seguir discutiremos as técnicas aplicadas em cada uma destas fases.

3.1 Estudo sobre sistemas semelhantes

Nesta fase foram analisados sistemas semelhantes ao Eugénio (7) e exploradas as diversas capacidades que estes ofereciam ao nível do desenvolvimento de novos teclados. Muitos sistemas não disponibilizam qualquer possibilidade de criação de novos teclados, outros dispõem de editores de teclados como o apresentado na Figura 3 e uma pequena minoria oferece um assistente para a realização desta tarefa [KeyVit03] (Figura 4). Nuns casos esta funcionalidade encontra-se incorporada no próprio sistema enquanto que noutros casos é implementada numa aplicação separada.

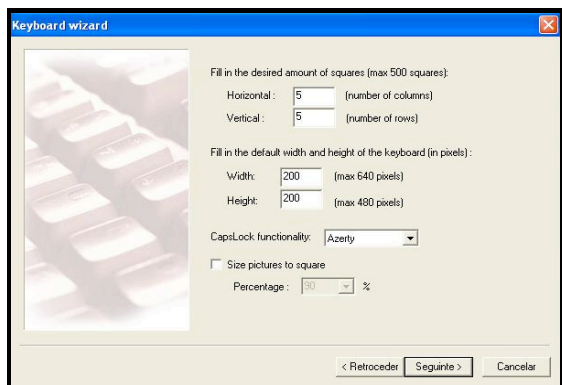


Figura 4 –Assistente de teclados do sistema KeyVit

A Joana decide criar um novo teclado específico para um novo aluno.
 Abre o editor de teclados, procura nos menus a opção Novo, que normalmente está no menu mais à esquerda com o nome Ficheiro.
 Clica em Novo e aparece uma caixa de diálogo onde insere o número de teclas do novo teclado e de seguida em OK.
 A aplicação apresenta uma "tabela" cujo número de células corresponde ao número de teclas inserida, cada célula corresponde a uma tecla.

Figura 5 –Excerto de um cenário textual

3.2 Caracterização dos utilizadores

Os utilizadores do sistema Eugénio são na sua maioria técnicos de reabilitação (e.g. terapeutas da fala, terapeutas ocupacionais, psicólogos), professores do 1º ciclo e professores do ensino especial. A área das tecnologias de reabilitação é recente e por isso a maioria dos cursos superiores que formam estes profissionais ainda não abordam de forma suficiente este domínio. Existem inclusivamente muitos profissionais em exercício que por terem obtido o seu grau há mais tempo não tiveram qualquer contacto ao nível do ensino superior com estas tecnologias.

Um estudo sobre a utilização das tecnologias da informação e comunicação pelos professores revela muitas destas dificuldades [Paiva02]. Entre outros aspectos refere que a grande maioria dos professores portugueses possui computador pessoal, mas que apenas aproximadamente metade o utiliza para realizar tarefas relacionadas com a escola e que a utilização do computador em actividades com os alunos é bastante modesta. Indica também que existe alguma utilização do computador em contexto educativo por parte de professores do primeiro ciclo, ainda que provavelmente, de uma forma algo incipiente.

Como apoio à caracterização dos utilizadores realizou-se também um questionário a vários técnicos de reabilitação e professores do ensino especial e regular (67), colocados em 14 instituições e escolas do distrito de Beja. A maioria eram professores do 1º ciclo (39%), do sexo feminino (79%) e com idade entre os 41-55 anos (49%). Neste questionário foram colocadas questões com o objectivo de se identificar o nível de conhecimentos dos utilizadores e os dispositivos e programas utilizados com mais frequência. O objectivo era por um lado identificar as competências dos utilizadores ao nível das tecnologias e por outro as características das interfaces mais difundidas junto destes profissionais.

A informação recolhida nesta fase foi incorporada no processo de desenho através do desenvolvimento de personagens, uma técnica que tem estado a tornar-se extremamente importante para a incorporação e partilha de conhecimento sobre os utilizadores [Thomaz05].

Como metodologia para a criação de personagens decidiu-se (1) analisar a informação obtida; (2) identificar vários tipos de utilizadores; (3) suportar cada um dos tipos de utilizadores com base em factos existentes nos dados recolhidos; (4) seleccionar os tipos de utilizadores mais verosímeis. Como resultado deste processo obtivemos três tipos de utilizadores materializáveis através de três personagens. Uma destas personagens foi baptizada de Maria, com 45 anos, Professora do 1º ciclo, sem qualquer formação ao nível das novas tecnologias e com uma utilização do computador essencialmente ao nível do processamento de texto. As outras personagens (Isabel, João) possuíam mais alguns conhecimentos ao nível da informática.

Para além do processo de construção de personagens ter obrigado a equipa a conhecer melhor as características dos utilizadores, o que por si só é um factor positivo desta técnica, a existência de utilizadores particulares também se mostrou útil na avaliação dos protótipos pois permitia confrontar as propostas de interface com as capacidades assumidas para as várias personagens.

3.3 Desenvolvimento de cenários de utilização

Com base no conhecimento sobre os utilizadores e na estrutura que descreve um teclado no sistema Eugénio [Garcia03] começaram-se a esboçar cenários de utilização do sistema. Numa primeira fase foram desenvolvidos cenários de utilização com base em

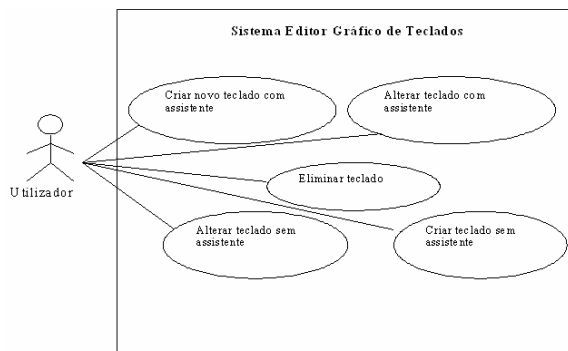


Figura 6 –Cenários de utilização do sistema

descrições textuais. Na Figura 5 apresenta-se um excerto de um destes cenários. Os cenários de utilização identificados nesta primeira fase encontram-se assinalados na Figura 6. No decorrer desta fase os cenários de utilização referentes à criação de teclado sem a ajuda de um assistente, e por isso mais flexíveis, foram abandonados pois os utilizadores do sistema não justificavam a sua existência. Os cenários definitivos foram então descritos de uma forma mais sistemática, na forma de sequência de acções, permitindo uma melhor identificação das trocas de informação entre o sistema e o utilizador.

3.4 Desenvolvimento e avaliação de protótipos de baixa fidelidade

Depois de estudados os vários cenários de utilização, foram desenvolvidos protótipos de baixa fidelidade da aplicação que foram avaliados em dois momentos. Num primeiro momento foram avaliados pelos vários membros da equipa de desenvolvimento que projectaram os protótipos num ecrã de grandes dimensões para a promoção da discussão entre os membros da equipa. Nesta avaliação tentaram-se determinar os pontos críticos de interacção, entre esta ou aquela personagem com a interface do sistema, na realização de determinadas tarefas. Depois de melhorados, os ecrãs dos protótipos foram impressos e avaliados junto dos utilizadores seguindo uma estratégia *Cooperative Evaluation* [Dix03]. Os utilizadores que participaram nesta avaliação foram três técnicos do Centro de Paralisia Cerebral de Beja ligados à equipa de aconselhamento de novas tecnologias (psicólogo, terapeuta ocupacional e técnico de informática), todos com conhecimentos ao nível da utilização do sistema Eugénio. Cada avaliação decorreu com a presença de apenas um técnico ao qual eram propostas tarefas que este tinha de realizar através do protótipo impresso. Para que o técnico pudesse indicar o que iria realizar em cada momento eram-lhe apresentados ecrãs do assistente. Sempre que o utilizador mostrava algumas dúvidas numa parte da interface, essa indicação era registada no próprio papel do protótipo. As dúvidas que normalmente surgiam diziam respeito à linguagem utilizada, assim como à funcionalidade oferecida por este

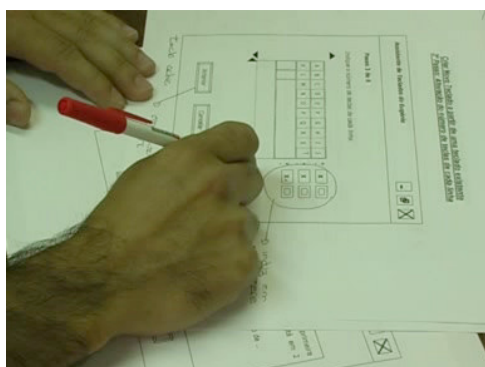


Figura 7 – Avaliação dos protótipos de baixa fidelidade

ou por aquele elemento. Estas avaliações também foram registadas em vídeo (Figura 7). Após a análise das notas efectuadas e dos vídeos de avaliação foram efectuadas novas melhorias no protótipo. Neste momento está a ser desenvolvido um protótipo de alta fidelidade da aplicação com recurso ao Microsoft Visual Studio 2003 codificado na linguagem de programação C#.

4. CONCLUSÃO E TRABALHO FUTURO

A aplicação de técnicas de desenho de sistemas interactivos centrados no utilizador (análise de tarefas, personagens, cenários de utilização e prototipagem) revelaram-se extremamente eficazes no desenvolvimento do sistema. Os resultados obtidos até ao momento indicam-nos que o recurso a assistentes para a configuração destes sistemas, e em particular o sistema Eugénio, é uma solução adequada para os técnicos da área, que para além de não possuírem muitos conhecimentos ao nível da informática têm de lidar com diverso software, cada qual com diferentes formas de configuração.

Após a sua conclusão o protótipo funcional irá ser testado junto dos utilizadores alvo.

5. REFERÊNCIAS

- [Dawe06] Dawe Melissa. Desperately Seeking Simplicity: How Young Adults with Cognitive Disabilities and Their Families Adopt Assistive Technologies. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Canada. 2006.
- [Dix03] Dix, Alan et al. Human-Computer Interaction. Prentice Hall. 2003
- [Garcia03] Garcia, Luis. Concepção, Implementação e Teste de um Sistema de Apoio à Comunicação Aumentativa e Alternativa para o Português Europeu. Tese de Mestrado. Instituto Superior Técnico. 2003.
- [KeyVit03] KeyVit On-Screen Keyboard. Technology and Integration. <http://www.tni.be/products/EurovocsSuite.php> (Junho de 2006)
- [MountFocus03] The MountFocus Keyboard Designer v. 3.2. 2003. MountFocus Information Systems Ltd. <http://www.virtual-keyboard.com/> (Junho de 2006)
- [Paiva02] Paiva, Jacinta. As Tecnologias de Informação e Comunicação: Utilização pelos professores. Departamento de Avaliação Prospectiva e Planeamento do Ministério da Educação (Programa Nónio XXI).2002.
- [PublicLaw86] Public Law 99-506. Amendments to the Rehabilitation Act of 1973. USA. 1986.
- [Thomaz05] Thomaz, Plinio. User Modeling with Personas. Proceedings of the 2005 Latin American conference on Human-computer Interaction. 2005.

ScWiz: Personalizing Psychotherapy

Marco de Sá Luís Carriço Luís Duarte David Cruz Cátia Torres
LaSIGE & Department of Informatics, Faculty of Sciences

University of Lisbon

Edifício C6, Campo Grande, Lisboa, Portugal

{marcosa,lmc}@di.fc.ul.pt, {lduarte,dcruz,ctorres}@lasige.di.fc.ul.pt

Abstract

This paper presents ScWiz, a framework that provides psychotherapists with means to tailor specific therapies for their patients. Used artifacts can be easily adapted to the patient's problem, to his evolution rhythm, therapy stage or even to possible filling in situations. Furthermore, the artifacts can be configured to pro actively adapt to the patient's behavior, according to previously defined rules, extending the therapist's motivating role. Besides it also enables deferred analysis of the patient's interaction and behavior with the therapy artifacts. The framework ranges from desktop computers, to mobile devices (e.g. PDAs and TabletPCs) covering the various therapeutic tasks for both patient and therapist. Special care was taken on the design stages, minimizing the intrusiveness of the solution and maximizing the simplicity of usage, particularly on the resulting patients' artifacts.

Keywords

Psychological diagnosis, self-therapy, PDAs, TabletPCs, patient/clinician interaction.

1. INTRODUCTION

Cognitive Behavioral Therapy (CBTs), on specific pathologies such as depression and anxiety may become a long, demanding and sometimes expensive process. Patients are driven, by the therapist, on a self learning and introspective process, replacing, whenever existing, their distorted cognitions with more adequate ones [12]. This requires a multitude of tasks to be performed by the patient and by the therapist, either cooperatively or individually. Therapists define procedures, questionnaires and the therapy itself before each session. On the other side, patients register life events and respective behaviors, schedule activities, etc, throughout the day. During face-to-face appointments, all the data gathered and produced between sessions is transferred between actors, orally or through paper artifacts. At this time, the therapist annotates the patient's evolution, conversation topics or even attitudes during the session.

Until recently, this process was supported mainly by paper or through some computer applications. However, due to the diversity of problems and associated therapies as well as the work flow inherent to the process, these solutions presented themselves as rigid, sometimes obstructing communication, therapy adaptation and even cooperation during sessions. This also caused delays, leading patients to frequently disengage from therapy. Based on this, a recent effort addressing these topics is providing new solutions and alternatives to previously used methods [4, 3].

For the therapist, the main problems to overcome are to:

(1) enable an easier creation or adaptation of the existing therapeutic material; (2) personalize therapy to the patient's specific needs; (3) create quick but structured annotations during sessions; (4) manage patient and session's information (5) a way to understand patient's difficulties with the therapy, outside the office and finally to (6) ensure that the patient's learning curve on the therapy process is short. As for patients, the main challenges are to provide means to easily fulfill their tasks, focusing on the treatment itself, and to ensure that motivation is kept during and between sessions, without the therapist's presence.

This paper presents **ScWiz**, a framework that supports the therapist's work, allowing him to tailor the therapy throughout the entire process. Integrated within the SCOPE architecture [5], it is also complemented by tools that allow therapists to perform deferred analysis, through the visualization of the patient's interaction with the therapy artifacts when apart from him. Furthermore, the resulting artifacts, configured by the therapist, enable patients to ubiquitously and quickly perform their tasks during their daily lives with aids and incentives, reducing effort and enhancing the therapeutic tasks. Special care was taken to provide therapists with enough power and functionalities to create diverse therapy artifacts but still maintaining usability and simplicity levels, on the resulting applications, very high.

It starts with a brief overview on current systems that relate to psychotherapy, followed by the description of the design process, the chosen architecture and an outline on available

features. Afterwards, the therapist framework **ScWiz**, this paper's main focus, and the patient's tool are presented. Finally, the evaluation that took place is addressed, some conclusions stated and future work is drawn.

2. RELATED WORK

Computer applications directed to psychotherapy have been gaining some momentum in recent years. As in many other areas on the health care domain, they often focus on data gathering or visualization, analysis and especially organizational tasks within the health care domain [8]. Specific software, directed to the psychiatric and psychological use, allows patients to follow particular methods of therapy and even diagnosis [14]. Excluding patient solutions that, relying on expedite approaches of diagnosis, have revealed strong human rejection [6], studies have demonstrated the effectiveness of the computer role in the process of anxiety and depression therapy [9, 19]. However, most of these systems provide either isolated therapist solutions or isolated patient solutions with no therapist control. Moreover, mostly rely on desktop approaches, which are incompatible with most of the previously described scenarios, including office consultation [11].

Several web-based self-help applications and websites are also emerging [1]. Overall, in spite of the common advantages, such as remote assistance and costs lowering, these techniques have many disadvantages as well [18]. Patient disengagement is frequent, as well as patient misinterpretation of the sites' objectives; sites require constant management and monitoring; and email and telephone assistance is often required.

More recently, hand held and overall mobile devices, such as PDAs or TabletPCs, became available, and new applications have appeared. However, they only cover partial steps of the therapy process and do not allow the customization of the patients' tasks or artifacts. The majority is rather simple and allows simple measurements of the severity of pathologies, indicates drug dosage or provides therapists with reference information about diseases or drugs [10]. On the patients' side, some self-control or relaxation procedures are available on hand-held devices [15]. Other applications developed for the treatment of specific pathologies, such as *bulimia nervosa*, also had positive outcomes [13]. But once again, these lack the possibility of adapting each step of the procedure or application to patients' specific needs.

Finally, none of the work found in the available literature addresses the therapy stages where the patient is away from the therapist, nor the problems that the patient faces during his homework.

3. SYSTEM DESIGN AND ARCHITECTURE

SCOPE is a project that addresses the aforementioned problems and aims to solve them [16]. The project followed a contextual design approach [2]. Whilst gathering the requirements interviews, with practicing therapists and researchers, were conducted, videos observed [7], sessions simulated and literature reviewed. During early stage

evaluation sessions, low-fidelity prototypes (for PDAs and TabletPCs) were created and used individually and collaboratively in simulated sessions by therapists and a group of varied background users [17]. Throughout this iterative process, users also participated and several sketches of possible user interfaces were drawn. These were repeatedly used and redesigned until high levels of user satisfaction were achieved.

The results of this process emphasized the need to use mobile devices, which ensure that the therapy artifacts can be carried along, used and (re)viewed by both therapist and patient whenever possible. Nonetheless, a larger platform should be included for the most complex features, for its larger computational power, storage space and working area.

3.1 Overall Architecture

Figure 1 depicts the general settings of SCOPE: one presumes a full resource environment (called fixed) that usually comprehends a PC, possibly a patient record server, and peripherals (e.g. printer); another (referred to as mobile) copes with the mobility requirement with variable resource capabilities and may be supported by a laptop or a hand held device; and the last covers the co-presence situations (named face-to-face), where TabletPC and hand held solutions are envisioned. The first two presume, primarily, an individual stand where therapist and patient work isolated from each other - collaboration through a network is also possible but it is the subject of JoinTS, a continuation of SCOPE.

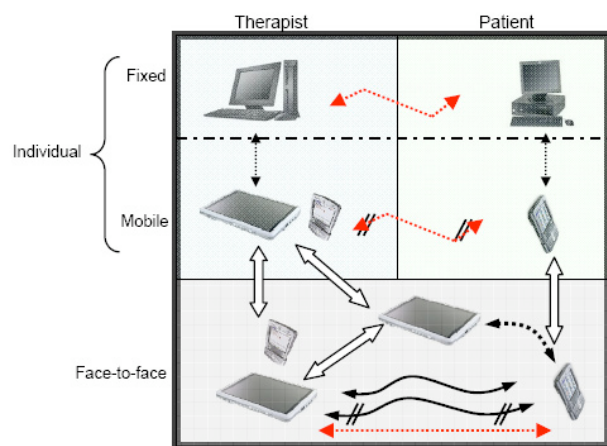


Figure 1. Global Architecture and usage scenarios.

On these settings, some scenarios can be envisioned that result in one or more applications. In general, each application may operate in stand alone mode, possibly integrating a classic paper based therapy procedure. On the other hand, applications can be articulated with each other, covering most of the therapy activities. Integration is accomplished through a shared database, exchanging artifacts and data through a synchronization process (represented by a strait-dashed line in the figure), or through di-

rect communication (a curly arrowed line). Note that direct communication also occurs between therapist and patient in the face-to-face setting. Applications can also be configured (depicted by a fat arrowed symbol) to become more adequate to a specific usage scenario.

3.2 Procedures and Features

Each procedure and feature is better suited to a specific scenario and correspondent platform. The following sections detail the usage scenarios and common tasks that are performed in each.

3.2.1 Therapy Definition

The therapy definition involves the selection and customization of forms; the creation of the items that compose it or even the analysis of gathered data. With SCOPE, using **ScWiz**, these tasks can be carried out ubiquitously, on a mobile device or at home/office, usually through a desktop computer. This latter platform provides therapists with a larger working area, particularly adequate to artifact customization practices or even analysis. On mobile settings, all the features are available. However, some of these are generally reduced to lighter versions due to the hardware's limitations. Still, synchronization between platforms is offered, allowing therapists to transfer data from one device to other, carrying only the necessary information for a specific day or session.

Depending on the used device's capabilities, information exchange can be done using a common connection between devices or Bluetooth. Alternatively, all the data can be exported to XML files and sent by e-mail or a digital card.

3.2.2 Therapy Adjustment and Collaborative Tasks

On face-to-face sessions, therapies and correspondent artifacts are usually adjusted instead of created, and tasks are mainly done in collaboration. On early stages of therapy, patients are usually guided and helped while completing forms or questionnaires, sharing a pencil and a paper or writing the same answers on separate paper sheets. SCOPE encompasses a similar scenario, where a shared device (e.g. Tablet PC) is used. When this cooperative mode is selected, all aids are disabled since direct intervention by the therapist is available.

When each actor has his own device, collaboration is done with wireless communication (Bluetooth and Wi-fi). Users interact with their own device and information is transferred between them as needed. However, this latter scenario also allows the therapist, when using a larger device (e.g. Tablet PC instead of a PDA), to monitor on his device the patient's application and actions, remotely disabling or enabling some functionalities. On these occasions, the therapist's interface is divided into a private space, where annotations can be taken and previous results analyzed; and into a public space where the patient's actions and screens are displayed.

3.2.3 Therapy Tasks

Therapy homework is usually done individually and away from the therapist. Mobile platforms are better suited for this type of therapy; however patients are also allowed to use their desktop computers for homework. However, this scenario is rarely applied since most of the tasks are to be completed during the day. Information exchange with therapists is also available through e-mail.

4 OVERALL FEATURES

To cope with the demands and complexity of the entire process of psychotherapy, including both session activities and homework, SCOPE contains some smaller applications that pertain each to a specific activity.

4.1 Annotations

Annotations are an important and recurrent activity during sessions. However, many times, the simple act of taking a note may retract from the conversation or task at hands. Preferably, these must be taken without distracting the patient but still with some detail and corresponding to the situation and action that was noteworthy.

SCOPE offers a set of utilities that cope with these difficulties. Annotations are taken on a mobile device just by choosing a theme and writing the note. They are automatically stored with the date of creation, the session number, the patient's name and with the chosen theme, for an easier recovery. On posterior analysis they can be (re)organized and new annotations can be attached to previous ones. Furthermore, recurrent themes, frequently used words are highlighted and emphasized if chosen by the therapist.

4.2 Analysis

Analysis is also possible for the remaining data handled by SCOPE (e.g. forms, questionnaires, thoughts, etc). On the desktop/laptop versions deep and thorough analysis is possible. Graphics and charts are automatically created for some questionnaires and hierarchic views of thought registers can be made. These forms of visualization help to emphasize patient's main problems or patterns of behavior. During analysis, categorization of data can also be done according to these results.

On mobile devices, a lighter version accommodates sufficient features to allow an easy search of the patient's previous results, common subjects within annotations or thought registers. These simple features are particularly suited to on-consultation settings where therapists can show patients details of their repetitive errors and frequent answers.

4.2.1 Deferred Analysis

Since a great part of the therapy process is accomplished away from the therapist, it was necessary to introduce a mechanism which allowed therapists to gain conscience on the patient's problems while working outside the consultations. To tackle this problem, a logging back end was introduced on the mobile applications. When a patient starts using an application, the log starts recording every action

until the patient exits the application. For each action that the patient takes (e.g. pressing a button, selecting an option or answering a question) a new log entry is added. Thus, the log is composed by a sequence of events that are recorded, including the type of event (e.g. main events or simple events) and the instant when the event was raised. However, the granularity of the events that are logged is also configurable by the therapist. Several levels are available ranging from logging each time the user taps the mobile device's screen or clicks the mouse button, storing the coordinates of such click, to higher level events such as when the user moves back or forward from one item to another within a form. This is particularly important when using mobile devices where the amount of memory is usually smaller.

The log files are stored in an XML like format and can be analyzed either manually or using the log player described on the following section.

4.2.2 Log Player

The log player is the application used to review the logs that were created whilst the patients used their applications. Typically, it presents a view of the application that the patient used and replays all the actions that the patient took. The amount of actions and events that are displayed depend on the granularity of the log that was recorded. However, when all the events are logged, the player also allows the user to increase the level of granularity, reviewing only the actions that are relevant for the patient at question. Furthermore, the speed in which the events are displayed is also configurable, particularly for large forms or questionnaires which contain up to hundreds of items.

5. TAILORING THE THERAPY

The artifacts that are commonly used on psychotherapy are based on groups of multiple-choice or free-text questions, items or fill-in fields. These items are grouped following pre-defined standards, composing, at times, extensive documents. Moreover, these are not always suited to every situation. For instance, a questionnaire containing questions regarding the patient's professional activity is not suited to an infant.

5.1 Personalized Artifacts

SCOPE offers therapists a way to overcome this problem. The **ScWiz** tool comprehends a set of mechanisms that permit therapists to personalize artifacts and correspondent applications to their patients. It integrates a database of standard used items, questions, etc; still allowing therapists to create any other item that is needed. Furthermore, it also supports the customization of each item's presentation (e.g. image, text) and interaction (e.g. multiple-choice item, free-text answer, gauge, drawing, etc). Creating a new questionnaire is resumed to simply selecting a set of items, to arrange them within a sequence and to define their above mentioned characteristics. This flexibility allows therapists do adapt the artifact, its presentation and interaction. For instance, when the patient is a child,

the form/questionnaire may be composed mainly by pictures with multiple-choice answers. On the other hand, for an adult, the same items may be textually presented and answered.

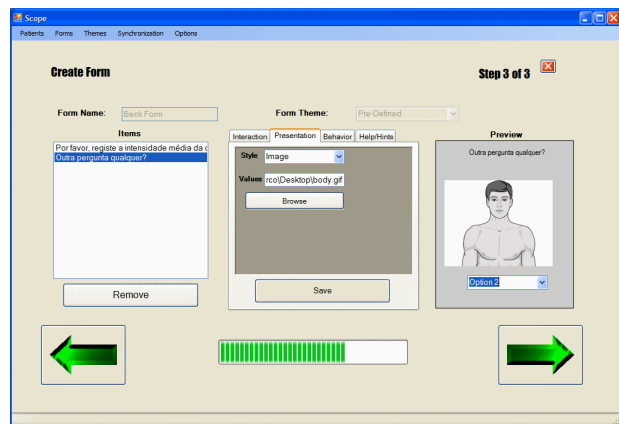


Figure 2. ScWiz: Form Creating Wizard

Another useful feature is to define questionnaires according to the place or time in which they will be completed. Imagine, for example, that a thought registering form must be completed during the patient's working hours. Here, time may be of utmost importance. Therefore, the therapist may define the artifact with hints or recurrent thoughts that can be easily chosen from a list, instead of written. On the other side, if the form is to be filled at night, then the patient is given space and freedom to write whatever he wants. The same methodology can be used to adapt a particular artifact to the patient's device. Smaller devices have limited capabilities regarding images or long answering lists. Furthermore, generally input possibilities are limited and data introduction should be facilitated. With **ScWiz** selecting adequate answering options and presentation types, the user interface is modeled taking into consideration all these dimensions.

It is also noteworthy, that default profiles are included and questionnaire patterns available to be chosen from wizards. To improve usability and promote a simple design and customization process, templates are provided, so that therapists can easily create either generic or more specific tools for their patients. Figure 2 depicts a wizard for creating an image based form. On the left side a list of the selected items is available. The box at the center presents the customization options (detailed in figure 3). Finally on the right, an automatically updated preview of the resulting application is shown. In this case, the form is completed with 3 steps: choosing the form name, the items and their presentation. However, the latter can be skipped and the default presentation is shown.

As shown in figure 3, several options to customize a resulting form or application are available. On the left, some of the interaction customizing possibilities are displayed. The first drop-box allows users to choose the type of interaction that will be used (e.g. drop-box, textbox, trackbar, etc). The remaining options are presented according to the

selection that was made on the interaction style dropdown. In this case, a list, representing the collection of items that will compose the drop-box, is shown. On the right side of figure 3, the help and hint configuration menu is presented. Here, therapists are required to select an operator, which, together with the value that is inserted, will trigger the help or hint that will be shown. Again, several options are available to define the presentation of the hint/help sentence or image. These are addressed with more detail on the following section.

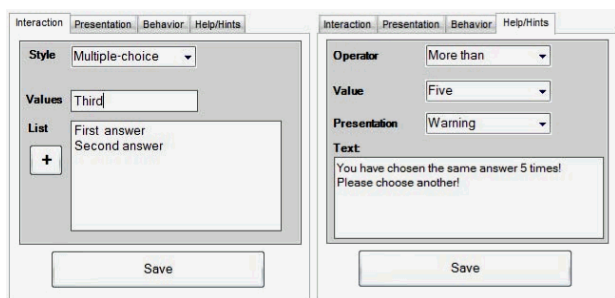


Figure 3. Customizing Features of ScWiz

Another advantage of such system is the ability to adjust these forms during sessions, on the mobile devices (Figure 4), according to just gathered information. Overall, forms, questionnaires, items and their composition can be matched to the patient's age, pathology, symptom, etc. These simple provide therapists with sufficient power to select the user interface of the patient's artifact.

5.2 Pro-active Motivation and Help

Through SCOPE, the therapist's work can be extended outside the office and sessions. For each item within a form, a number of answering aids can be provided (Figure 4). These can also assume various presentation and interaction forms. The therapist, while creating a new artifact or updating an existing one, is able to choose the schedule, type and amount of help that will be provided to the patient (Figure 4). Generally, each type of item and corresponding interaction and presentation is associated with a type of hint (e.g. image presentation uses a new picture to help). However, new combinations can be made, switching either the content of the hint message or even the way it is displayed. Again, details such as the location and time of completion can be anticipated, thus optimizing the artifact according to it.

Patient's motivation is also kept when away from therapy sessions. Using similar methods, the therapist defines motivating comments, incentives and congratulations when an item is correctly answered or even when a certain amount of thoughts is registered during a day. To activate these rewards, the therapist defines rules for each item within a form. Each rule is composed by a message (e.g. hint, warning or motivating message), the presentation and interaction types, the trigger and the frequency with which it should be presented. However, rules that entangle various items are also possible. For instance, if the patient

answers correctly to the first five questions, then hints are automatically disabled. On the other hand, if the patient fails to answer correctly the five following questions, then hints are once again activated. Details such as the amount of positive or negative results are also configurable.

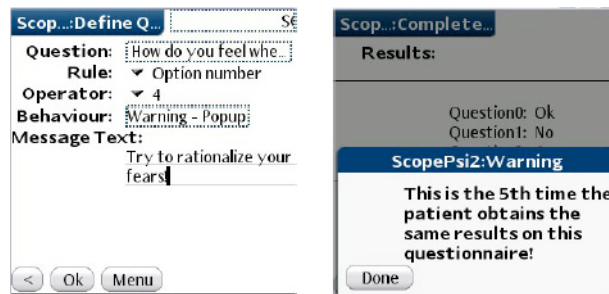


Figure 4. Defining rules on a Palm Device

An alarm mechanism assures that patients are reminded to complete their tasks during the day. Once again, the frequency or time of day of such occurrences is also defined by the therapist. However, the therapist is also allowed to define criteria that force the alarm to repeat more frequently if the patient fails to perform one or more tasks that were scheduled.

6 THERAPY ENDEAVOR

The patient's tool allows patients to engage on therapy as previously defined by the therapist. It reads each created specification materializing it into a pro-active artifact.

6.1 Using the Artifacts

Each task that is usually done during therapy is replicated and enhanced so the effort in accomplishing it is diminished and the focus directed to the content that needs to be provided. For instance, multiple-choice questionnaires are easily filled by selecting the correct answer. However, browsing the questionnaire forth and backwards allows patient's to update questions at any given time without creating confusing or unreadable documents. Furthermore, each of these updates is stored and provided for later analysis. On thought registers, specific fields are generally provided (e.g. situation, emotion, behavior, etc) and hierarchically presented in order to facilitate the filling-in process.

Activity scheduling is also improved since recurrent activities are available for the patient to choose and the day is divided into several slots for an easier organization. On this aspect, later classification of each activity is also possible.

During each of these tasks, the application adapts to the patient's behavior, according to the rules that were defined by the therapist.

6.2 Therapy Tutorials

Also included on the patient's tool is a customizable tutorial on several techniques and common procedures (e.g. relaxation, etc). Patients that are selected to be enrolled in group therapy can study previously and on a regular basis

the main activities that will be undertaken. Alarms are generally activated for patients on early stages of group therapy whereas for those who are more experienced, alarms are only sounded when an appointed task is missed. As for the therapy artifacts, tutorials can also be presented in different ways. Suggestions and hints can also be popped up during browsing, if defined so by the therapist.

7 EVALUATION

The therapist framework and corresponding patient's tool are being used on a clinical and on an academic scenario. However, these were firstly evaluated during two months by several users with different backgrounds.

7.1 Usability Evaluation

The therapist's framework was tested by a group of three practicing therapists and two researchers. The main focus of this group was to test the feasibility and effectiveness of the artifact creation and customization feature. Analysis and respective features were also assessed during this stage. On a later stage, therapists also tested the therapy application. However, a larger group of users from various backgrounds was responsible for carrying a mobile device with the therapy tool for several days. Various tasks were scheduled and completed during this period. Some sessions were also simulated with the various devices (Figure 5). All of these sessions were filmed for posterior analysis. During these tests some measurements were made.

7.1.1 Results

Again, the mobile devices proved to be adequate on the various settings that were tested (e.g. individual, consultations). Therapists were required to create three different artifacts (e.g. questionnaires, tests) composed by different types of questions and items. Once again, different subjects were involved. Therapists also responded to the created artifacts, took some annotations and transferred the information between devices. In order to assess the system's advantages during consultation activities, some users were asked to accomplish the same tasks in the traditional way. The time spent in accomplishing each set of tasks was measured.

Users took approximately 1 minute to create each item, choosing its hints, behavior, interaction element and content. To create a 10 item questionnaire (after the items were created), the time spent was about 2 minutes. The users took from 2 to 4 minutes to complete the artifacts, depending on the amount of free-text questions on the artifact. Comparing to the time used for the same tasks without the system, SCOPE proved to be an advantage for both therapists and patients. The major benefits were noticeable when creating the artifacts with differences ranging from 50 to 80% less time using SCOPE.

Overall, results demonstrate that the **ScWiz** tool can be used effectively for artifact creation and content organization. Therapists welcomed the ability to configure the interaction elements and choose the questions/answer types. For specific therapies, therapists were particularly interested in the possibility of providing help and hints to their

patients. Globally, the integration of the several activities within the same tool was positively received by all the users.

Regarding the analysis application different opinions emerged. Some of the therapists thought it should allow more complex visualization of data, namely different types of graphic charts. The rest preferred the tool as it is, since the output is easily interpreted both by them and by patients, stating that if more power was needed a desktop or TabletPC could be used. The users who tested the patient application appreciated the possibility of completing questionnaires using the hand-held devices, particularly because they can browse through the already answered questions and update them if necessary. Another major benefit was the immediate feedback that the tool provides, if configured to do so, by the therapist.



Figure 5. Evaluation session with mobile devices

7.2 Resulting Applications' Evaluation

To validate the expectations regarding the customization capabilities of **ScWiz** a second evaluation stage took place. The main goal during this stage was to replicate and enhance existing and successfully used psychotherapy applications. Accordingly, these tasks were accomplished by therapists, some with previous experience in working with such software. However, now, they were given the capability to manage and adjust the applications to their specific needs.

As expected, some rearrangements were necessary to accommodate all the applications that were emulated. However, these were successfully accomplished and are now integrated within the current version of the software.

7.2.1 Anxiety

Most of the available applications which address anxiety problems rely on relaxation tutorials or anxiety assessment inventories [15]. With **ScWiz**, therapists were able to adjust tutorials for similar procedures but further enhancing them. Among others, they were able to include exemplifying images, hints on how to execute the movements as

well as a classifying field on the effectiveness of the results. Some of these methods were previously available on the usual paper based artifacts while others were here introduced since the used medium provided new possibilities. The used images were previously owned by the therapists or were quickly downloaded from the web.

For the anxiety inventories, therapists selected a simple arrangement of a few items and the correspondent levels of intensity for each (e.g. Beck Anxiety Inventory, Max Hamilton Test). These were particularly appreciated by final users who tested the therapy application. The main reason was the simplicity, quicker response time and possibility to update each item at any time, especially when compared with the traditional paper form. For the therapists that participated on these evaluation sessions, these results were extremely important, since the ease of use is essential for the success of the therapy process. Patients frequently disengage from therapy due to the effort involved in completing all their tasks.

7.2.2 Depression

Pleasant Activity Scheduling is a common practice during depression therapy. Using **ScWiz**, therapists were also able to compose a simple scheduling form which integrated classification fields for each task that was scheduled (Figure 6). Hints were added mainly as lists of common activities that could be selected for a slot within a day (Figure 6, right). Furthermore, the system includes a mechanism that memorizes recurrent activities that are introduced by the patient. These are also provided through lists, minimizing the need to write on the device.

Alarms were also defined for these forms so that final users were alerted when tasks needed to be scheduled. A set of different arrangements resulted from this stage. For instance, one of the therapists composed a scheduling form where a patient could browse each hour of the day sequentially. For each hour an activity could be planned and an annotation could be attached to it, including information such as the location of the activity or any other data that the patient might find useful. On the other hand, another resulting form, allowed patients to introduce the date (e.g. hour) and choose an activity for that slot. According to the therapist, this form was particularly adequate to latter stages of therapy, where the patient is given freedom to plan his activities as wished. However, despite the different arrangements that distinguish these forms, both could be re-visited and updated, registering also the pleasure and efficiency with which the activity was complete.

7.2.3 Pain and Weight Disorders

For the treatment of pain associated disorders, therapists created image based forms. One of the resulting forms displays body images where users point the location of the pain directly on the image (Figure 7, left). Associated to each image is a set of questions related to pain intensity, used medication or associated symptoms. Generally, each item pairs an image with a trackbar, allowing the patient to quantify his pain. Once again, hints and triggers were

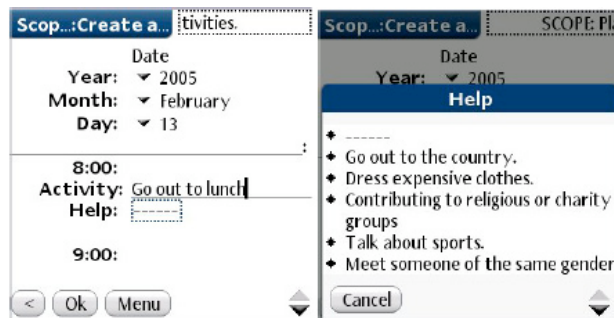


Figure 6. Activity Scheduling on a PalmOS PDA

defined. Whenever patients choose a high level of pain intensity, relaxation sentences and hints are displayed. On certain occasions, instructions on how to react to the symptoms were also defined.

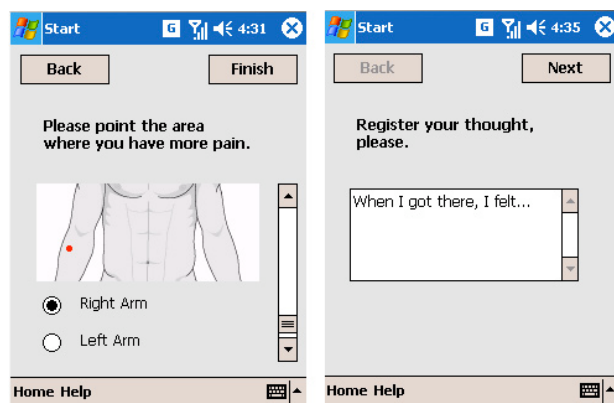


Figure 7. Pain Therapy on a Pocket PC

Some forms addressing weight associated problems were also created. These were mainly questionnaire based but some also incorporated a thought registration component (Figure 7, right). Regarding this latter facet; it was easily incorporated and achieved through a simple arrangement of a few items (e.g. situation, consequent thoughts, behavior and adequate thought).

8 CURRENT AND FUTURE WORK

The framework and resulting tools are being used on a clinical environment where new usages are already being discovered. The achieved flexibility of **ScWiz** allows therapists to replicate previously successfully used tools, to extend them and adjust them to the same pathologies to which they were designed. Initially intended to support anxiety and depression, it was quickly extended to other pathologies. Weight disorders and pain treatment are examples of different applications that SCOPE is able to accommodate.

Following the positive results that are being achieved with SCOPE, a new project that deals with group therapy sessions is already being developed. Besides providing individual support for each patient, this newer version provides

streams of communication between (1) patients within a group therapy session; (2) therapists also within a group therapy session; and (3) between all the patients and the therapist. Finally, a (4) shared space where both patients and therapists can post their comments is also available through an interactive smart board. Outside sessions, smart phones will allow therapists to be notified of patients' accomplishments or even permit distributed motivation through group members.

The use of new equipment will also provide new functionalities to the system. Sensors (e.g. temperature, heart rate, etc) will be incorporated so that patients with severe situations can be monitored. If necessary, their therapists might even be notified through an automatic SMS message. Also, using smartphones with built-in cameras, new techniques for weight disorders can be used as well.

9 CONCLUSIONS

Recently, some attempts to support psychotherapy with the use of computers have been made. These are available either through the internet, desktop computers, and lately even on mobile devices such as PDAs. Desktop approaches are clearly inadequate to the ubiquitous needs of therapy tasks and can be intrusive during consultation settings. When it comes to mobile applications, these are too specific, either covering only partial steps of the process or specific pathologies. Moreover, therapist intervention and management over the therapy is never offered, particularly when away from the patient.

This paper presents a novel approach which addresses the majority of these problems. It includes a wizard based tool, which allows therapists to compose specific artifacts to each patient and his needs. Furthermore, with the resulting applications, patients can engage on their therapeutic tasks whenever and wherever they are, using a less time consuming and demanding tool than the common passive paper approach. The extension of the therapists' motivating and educational role is also accomplished by the presentation of hints, aids and incentives during form and task completion.

10. ACKNOWLEDGEMENTS

This work was partially funded by FCT (Fundação para Ciência e Tecnologia) through project JoinTS and LaSIGE (Large Scale Informatic Systems).

References

- [1] Gerhard Andersson and Viktor Kaldö. Internet-based cognitive behavioural therapy for tinnitus. *Journal of Clinical Psychology*, 60(2):171–178, 2004.
- [2] H. Beyer and K. Holtzblatt. *Contextual Design: A Customer Centered Approach to Systems Design*. Academic Press, San Francisco, CA, USA, 1998.
- [3] Luís Carriço and Marco Sá. Hand-held psychotherapy artifacts. In *Proceedings of the 11th Human Computer Interaction International Conference, HCI05, Las Vegas, USA, 2005*.
- [4] Luís Carriço and Marco Sá. Mobile devices for active psychotherapy. In *Proceedings of Applied Computing, International Association for Development of the Information Society, IADIS, San Sebastian, Spain, 2006*.
- [5] Luís Carriço and Marco Sá. Proactive psychotherapy with handheld devices. In *Proceedings of the 8th International Conference on Enterprise Information Systems, ICEIS2006, Paphos, Cyprus, 2006*.
- [6] Amar K. Das. Computers in psychiatry: A review of past programs and an analysis of historical trends. *Psychiatry Quarterly*, 73(4), 2002.
- [7] Joan Davidson, Jacqueline B. Persons, and Michael A. Tompkins. Cognitive-behavior therapy for depression, 2000.
- [8] C. S. Garrard. Human-computer interactions: can computers improve the way doctors work? *Schweitz Med Wochenschr*, 130:1557–63, 2000.
- [9] Lina Gega, Isaac Marks, and David Mataix-Cols. Computer-aided cbt self-help for anxiety and depressive disorders: Experience of a london clinic and future directions. *Journal of Clinical Psychology*, 60(2):147–157, 2004.
- [10] Michael A. Grasso. Clinical applications of hand held. In *Computing. Proc. Of the 17th IEEE Symposium on Computer Based Medical Systems*, 2004.
- [11] Paul Luff and Christian Heath. Mobility in collaboration. In *Computer Supported Cooperative Work*. ACM Press, 1998.
- [12] M. Mahoney. *Constructive Psychotherapy*. The Guilford Press, New York, 2003.
- [13] Margo Norton, Stephen A. Wonderlich, Tricia Myers, James E. Mitchell, and Ross D. Crosby. The use of palmtop computers in the treatment of bulimia nervosa. *European Eating Disorders Review*, 11(3):231–242, 2003.
- [14] Judith G. Proudfoot. Computer-based treatment for anxiety and depression: is it feasible? is it effective? *Neuroscience and Biobehavioral Reviews*, 28:353–363, 2004.
- [15] Amy Przeworski and Michelle G. Newman. Palmtop computer-assisted group therapy for social phobia. *Journal of Clinical Psychology*, 60(2):179–188, 2004.
- [16] Marco Sá and Luís Carriço. Psychological therapy artifacts - usage and editing with pdas. In *Proceedings of Applied Computing, International Association for Development of the Information Society, IADIS, Algarve, Portugal, 2005*.

- [17] Marco Sá and Luís Carriço. Low-fi prototyping for mobile devices. In *Extended Abstracts of CHI2006*, ACM, Montreal, Canada, 2006.
- [18] Deborah F. Tate and Marion F. Zabinski. Computer and internet applications for psychological treatment: Update for clinicians. *Journal of Clinical Psychology*, 60(2), 2004.
- [19] J. H. Wright and A. Wright. Computer-assisted psychotherapy. *Journal of Psychotherapy Practice Research*, 6:315–319, 1997.

Supporting Group Analysis in Psychotherapy

Luís Carriço Marco de Sá Luís Duarte Tiago Reis
LaSIGE & Department of Informatics, Faculty of Sciences
University of Lisbon

Edifício C6 Campo Grande, Lisboa, Portugal

{lmc, marcosa}@di.fc.ul.pt, {lduarte, treis}@lasige.di.fc.ul.pt

Abstract

Psychological therapy for pathologies such as depression, anxiety, among others, often leads patients to group therapy sessions, where patients and therapists interact and share their experiences. Besides including the challenges that are present in any kind of meeting settings, psychotherapy, for its critical health scenario, presents difficulties that have not yet been approached by currently used groupware. This paper introduces a system, which uses mobile devices and a large public screen, to support group therapy sessions. It encompasses a variety of functionalities that provide both therapists and patients with necessary means to enhance and fulfil their particular tasks during and between sessions.

Keywords

CSCW, Psychotherapy, Mobile Devices, Groupware, Smart Boards.

1. INTRODUCTION

Psychotherapy, on its various forms and orientations, requires interaction between patient and therapist. Both actors engage on sequential meetings where information is exchanged orally, through conversations, and through paper artifacts. These artifacts are completed by both, cooperatively, or by the patient, during the meetings, at home as homework or during the daily life, registering important events that matter to him [5].

Group therapy assumes all the above mentioned activities and introduces new ones. Here, global procedures are maintained, but the number of participants, information and used artifacts is larger, increasing the challenges for both therapists and patients. Furthermore, group therapy sessions are many times guided by two therapists, introducing a new communication channel. However, conversations and information exchange between therapists are usually reduced to pre or post-meeting sessions and interaction during session is avoided. Cooperation is limited and generally one therapist assumes a passive role, observing the meeting and taking notes. The management of patients' information is difficult and at times obstructs therapists from accessing necessary data during sessions. For patients, the completion of tasks, using the usual paper forms and questionnaires is demanding and time consuming. Furthermore, patients are only able to share their experiences orally, and homework and completed artifacts are rarely (re)viewed by everyone.

Nowadays, these challenges are yet to be addressed by specific groupware and cooperative work systems, which are suited to these critical scenarios. Overall, it is paramount to provide therapists with: (1) a way to access necessary information without interrupting the sessions; (2) support the information flow between therapists

without annoying patients; (3) provide means to interact with all patients equally, sharing information between all. On the other hand, for patients the main problems to be solved are the access to the used artifacts, their completion and filling-in and a way to easily share their work with their colleagues.

This paper presents the JoinTS (Joint psychological Therapy Support) system, comprised by mobile devices and a public display (e.g. smart board), which addresses the aforementioned problems and aims to enhance the group therapy process. The paper starts with an overview of the project's main goals, followed by the description of the envisioned solution's architecture. Afterwards, the features that tackle each of the procedures and respective problems that pertain to the group sessions are described. Finally some related work is presented, conclusions are discussed and future work is drawn.

2. JOINTS

JoinTS is a project that aims to provide adequate computational support to group therapy, in several scenarios, with different hardware and software configurations and covering a multitude of psychotherapy activities. The JoinTS's system covers from classical group therapy scenarios, in one room with the presence of all participants, to completely distributed scenarios (e.g. "holiday" groups), through spontaneous gatherings (e.g. "coffee-table" scenarios), combinations of the above, etc. Some of these are "tentative" situations that emerge from the introduction of the technology, but need clinical validation.

Hardware configurations also vary. A single display groupware setting (large public display - LPD) seems adequate, for instance, to the classical scenario, either

providing direct interaction with an interactive LPD or indirect through the therapist device. Multiple participant interaction with the LPD is also envisaged, although most of the times, the therapist control seems clinically preferable. Participants and patients in particular may use PDAs, small emerging tablets, smart-phones or simply mobile phones. The main therapist most adequate devices should be TabletPCs, but PDAs and even phones are admissible particularly in conjunction with an interactive LPD. Auxiliary therapists' may recur to PCs. In other scenarios, the LPD may not be possible or multiple LPDs might be needed (e.g. two room sessions).

This paper addresses the LPD setting. Most notably, it focuses the analysis activities where the tasks performed by patients, either at home or during the session, are presented and discussed.

2.1 Therapy Analysis LPD Architecture

Figure 1 depicts the LPD architecture. The scenario encompasses a public screen (the LPD) and corresponding application, a session server, a session facilitation tool, patient applications and, eventually, an auxiliary analysis tool.

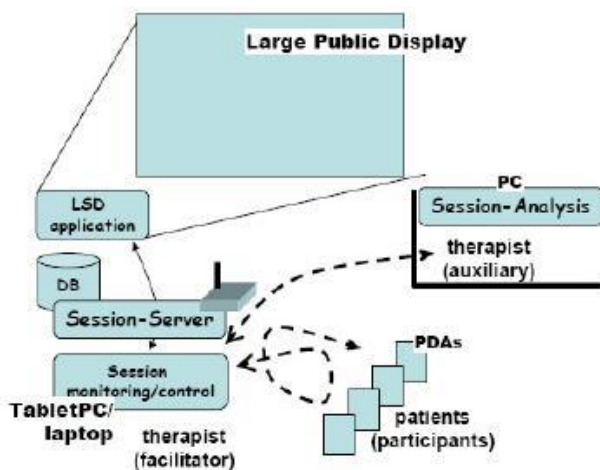


Figure 1. Architecture for the Large Public Display Setting

The session-server is the central component of the architecture. It provides access to a storage space (DB in the figure), that keeps records about patients, their work and historical progress during therapy and info about groups and sessions. Above all, the session-server is responsible to ensure the flow of communication during sessions, including the information that is displayed in the LPD. The session-server usually runs on a dedicated PC in the session room.

The LPD application shows the information in the LPD. For interactive LPDs it is also responsible to forward modifications to the server. It is up to the server to maintain the coherence between the information shown in the LPD and other local representations of that display (e.g. session facilitation). The LPD application usually runs on

the server PC, although remote settings are also envisaged (e.g. a two-room setting).

The patient tools are form-filling tools (e.g. questionnaires answering, activity and thoughts registration), running on a PDA/phone based platform. Patients transfer entered data to the server as needed (at the beginning or during the session) and it is up to the server to notify the other tools (e.g. session facilitation). Patients also receive forms from the server. Patient tools of this scenario are the same used on individual therapy (see [2] for details).

The session facilitation tool enables therapists to analyse and monitor patients' data, communicate with the auxiliary therapists and patients, manage what information will be shown in the LPD and in general control all the information exchanged in sessions. It usually runs on a TabletPC held by the main therapist. Situations may arise where the TabletPC is also the server PC, running the LPD application on a second monitor output.

The session analysis tool is a subset of the monitoring/control, disabling the management of the LPD space and the communication with the patients. A wireless, access-point based, network ensures the connection between tools. Other configurations are possible but more adequate for alternative scenarios. XML is used in all exchanged messages. The server receives messages from the clients, checks the destination, saves them in the DB as needed and forwards to the correct user or users.

3. THE SESSION FACILITATION/ANALYSIS TOOL

Figure 2 shows the facilitation tool. Three workspaces are identified, corresponding to the three available "tabs". The first one ("my space") corresponds to the therapist private space. The second one ("shared space") corresponds to the workspace shared with other therapist. The third one ("display space"), only available for the facilitator, corresponds to the public display space.

The private space, selected in the figure, is used to prepare and analyse information. The visible information reveals pain therapy questionnaires from three different patients. The shown objects are exact replicas of the corresponding patient-PDA artefacts, and may correspond to a home work or to an ongoing patient task. In both cases, the buttons on the top of each artefact enable the therapist to browse each patient's data independently. The ones next to the tabs permit to advance all forms simultaneously. In the case of an ongoing patient task, the artefacts are (optionally) updated as patients interact with their own PDA. This allows the therapist to monitor the patients' activities, including their difficulty to interact with the PDA application.

Once the therapist is satisfied, it may send the information to the other therapist or publish it on the LPD just by dragging all or a subset of the patients' artefacts to the corresponding tab selector. In both cases, the information is duplicated in the corresponding tabs of the local tool, and sent, through the server, to the remote application (the other therapist tool or the LPD application).



Figure 2. Workspace Management

On the right side of figure 2 two vertical bars are visible that provide access to a set of auxiliary components. When opened they offer the elements visible in figure 3. The "Listing tool" (on the left) is used for the selection of data, usually published to the private space. It permits several working modes. For example: (1) select data for one specific form, from all selected patients; (2) select data of several forms for one patient (view patient evolution); (3) select ongoing patient task. Specific patients can also be selected and the destiny workspace chosen.



Figure 3. Session definition & auxiliary channels

On the right of figure 3 two other components are available. At the top, a chat tool enables an instant messaging service between therapists, usually from the auxiliary to the facilitator. In conjunction with the therapists' "shared space", provide an important, non-disturbing, means of communication between therapists, not possible without in paper based-sessions. At the bottom an annotation tool is available.

4. RELATED WORK

In order to fully understand a collaborative system, it is necessary to study two basic aspects of its workflow: how users perform their individual work and how they relate it in a collaborative way [7].

Group therapy scenarios often rely heavily on communication in order to achieve the purposes of the system. Pinelle [3] points a few key factors for the construction of a successful collaborative system: support of explicit communication - both spoken and written as the first type is the most commonly used in real collaborative environments and the latter is crucial for questionnaire filling during therapy sessions. Another important aspect in groupware is related to shared-workspace usage. By providing a space where each intervenient is able to see what another is doing, it is possible to reduce communication traffic in the system. Although this may be seen as a major advantage in a system, it also has major drawbacks: it is necessary to manage and protect each user's work in the shared-workspace to avoid conflicts and collisions during work session. The final issue presented is related to artifact sharing in collaborative environments. Traditionally, one would have to physically deliver an artifact to another in order to share it. In groupware applications it is possible to perform this action in an asynchronous way, through wireless communication.

The Pebbles project [6] was an academic research whose main goal was to connect several hand held devices to a PC. The portable devices would interact with the main computer in several applications, which allowed them to control the keyboard individually. The system also allowed an individual to see what his colleagues were doing by displaying overall information on a whiteboard. Still, the management of different groups with specific needs is scarcely manageable. Furthermore, the specificity of the group therapy scenario requires details such as the easiness of use and the specificity of the used artifacts, particularly for the patients' client applications, to be of utmost importance.

Another interesting system is Teamspace [4], jointly developed by Boeing and IBM. The project's goal was to support cooperative teamwork in a spatially distributed environment. The main focus was to support task scheduling in the system, managing workflow and sharing work artefacts across a significant time period. The system also allowed for synchronous meetings in which all intervenients would review each other's work and schedule new tasks. However, the online monitoring of several working spaces is not included in the system, whereas for psychotherapy, it is essential for therapists to have the possibility to be aware of every patient's action.

Overall, despite their technological potentialities and wide roll of features, current collaborative systems and the several modules that support it, can not be directly used on the group therapy setting. Besides the critical scenario and the particular settings in which these take place, the needs of both therapists and patients need to be taken into consideration and must be included in a single

system. Furthermore, the system must also support the management of information that might be generated by several working teams and patient groups. Another important aspect is the continuous nature of the therapy process, where the pre and post meeting work, either isolated or done in cooperation, is essential to the success of each meeting. Accordingly these tools, that support off-session work, must comply with the needs and features that are incorporated into the overall architecture.

5. CONCLUSIONS AND FUTURE WORK

Current group psychotherapy procedures and used materials introduce challenges to the therapy and collaborative process that result in demanding and time consuming activities for all the actors, at times, even preventing the achievement of better results. Most importantly, these challenges can not be solved by generic groupware tools or by existing psychotherapy software.

This paper presented JoinTS, a project which addresses this problems and aims to support group therapy sessions, integrating individual therapy solutions [1], enhancing the work flow and communication during sessions, but extending its scope through various settings. The adopted solutions take advantage of the emerging mobile technologies, which provide the mobility and necessary characteristics to be used on several scenarios, without disrupting and enhancing the currently used processes. A large shared display is also included to facilitate the dissemination of information between users and to support the management of each session as well as the analysis and visualization of multiple data from the various members of the group.

Current and future work includes new visualization patterns and templates, particularly adequate to share results, patients' evolutionary paths and statistics with therapy groups. On this facet, the monitoring functionality is being extended in order to allow a deferred visualization of the patients' behaviour. Besides supporting the revision

of group sessions, it also supports the deferred analysis of the patient's behaviour while working away from the therapist.

6. ACKNOWLEDGEMENTS

This work was partially funded by FCT (Fundação para Ciência e Tecnologia) through project JoinTS and LaSIGE (Large Scale Informatic Systems Laboratory).

7. REFERENCES

- [1] Luís Carriço and Marco Sá. Hand-held psychotherapy artifacts. In Proceedings of the 11th Human Computer Interaction International Conference, HCI05, Las Vegas, USA, 2005.
- [2] Luís Carriço and Marco Sá. Proactive psychotherapy with handheld devices. In Proceedings of the 8th International Conference on Enterprise Information Systems, ICEIS2006, Paphos, Cyprus, 2006.
- [3] D. Pinelle et al. Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. *ACM Transactions on Human Computer Interaction*, 10(4), December 2003, 281-311.
- [4] W. Geyer et al. A team collaboration space supporting capture and access of virtual meetings. In Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, 2001.
- [5] M. Mahoney. *Constructive Psychotherapy*. The Guilford Press, New York, 2003.
- [6] Brad A. Myers. The pebbles project: Using pcs and hand-held computers together. In CHI '00 Extended Abstracts, 2000.
- [7] C.G. Wolf and J. Karat. Capturing what is needed in multi-user design: Observations from the design of three healthcare systems. In Proceedings of DIS, Amsterdam., 1997.

Dispositivos de Gestão da Atenção em Sistemas Colaborativos

António Ferreira Pedro Antunes
Faculdade de Ciências da Universidade de Lisboa
{asfe,paa}@di.fc.ul.pt

Sumário

*Neste artigo apresentamos uma perspectiva combinada sobre Attentive User Interfaces e sistemas colaborativos para ajudar a resolver o problema de como atenuar a distração decorrente das múltiplas interrupções que caracterizam a colaboração mediada por computador, mantendo simultaneamente a necessária noção de consciência do grupo. Propomos uma abordagem para o estudo da atenção do grupo organizada em fluxos de informação e em dispositivos de atenção. Com base em factores psicológicos conhecidos, propomos quatro destes dispositivos, que são: o *time separator*, o *opportunity seeker*, o *change emphasizer*, e o *activity anticipator*. No final aplicamos a abordagem ao desenvolvimento de uma ferramenta de brainstorming electrónico.*

Palavras-chave

Attentive User Interfaces, sistemas colaborativos, fluxos de informação, dispositivos de atenção.

1. MOTIVAÇÃO E TRABALHO RELACIONADO

Desde finais dos anos 90 que vários investigadores da área da Interação Pessoa-Máquina (IPM) têm mostrado interesse nas potencialidades das Attentive User Interfaces (AUI), motivados pelo reconhecimento de que, à medida que as necessidades de informação e comunicação aumentam, também sobem os custos das interrupções e das faltas de atenção [Vertegaal03a]. Assim, em vez de assumirem que o utilizador está sempre atento e disponível para gerir e otimizar a sua atenção (ex. mostrando ou ocultando janelas no ecrã), as AUI negociam, em vez de imporem, a atenção dos utilizadores através do estabelecimento de prioridades para apresentar a informação.

A maior parte da investigação em AUI está direccionada para actividades realizadas por um só utilizador, sendo a principal assumpção que o desempenho individual diminui com o aumento do número de pedidos simultâneos de atenção. As experiências realizadas nesta área tipicamente estudam os efeitos numa tarefa primária causados por interrupções que exigem o processamento de uma tarefa secundária, possivelmente não relacionada e inesperada, como estar a editar um programa e, ocasionalmente, realizar mentalmente uma multiplicação [Fogarty05b].

Com base nestes pressupostos, vários investigadores estão a melhorar tanto os dispositivos de entrada como os de saída para que o utilizador permaneça focado na tarefa primária sem se distrair demasiado com a tarefa secundária. Isto pode ser conseguido utilizando sensores que detectam o estado de atenção transmitido pelo olhar ou postura corporal do utilizador (entre outras possibilidades) [Vertegaal03a, Fogarty05a], utilizando modelos estatísticos de interrupção que determinam a melhor ocasião para comunicar com o utilizador [Horvitz03, Fogarty05b], ou gerindo o nível de detalhe da apresentação de acordo com o foco de atenção do utilizador [Vertegaal03a].

Em paralelo com os avanços das AUI, na área dos sistemas colaborativos suportados por computador, tem-se verificado a adopção de técnicas que procuram melhorar a consciência do grupo (uma propriedade relacionada com o sentido de proximidade dentro do grupo) baseadas na partilha de *cada vez mais* informação sobre as acções que os membros do grupo realizam. Um dos argumentos apresentados é os canais de comunicação mediados por computador serem relativamente pobres quando comparados com meios mais naturais, como a interação face-a-face [Gutwin02]. Contudo, um problema desta tendência é não reconhecer que *mais pode significar menos* devido às limitações da capacidade de atenção humana, especialmente quando as pessoas ficam rodeadas de computadores e de informação (não necessariamente útil).

Existe alguma investigação sobre a aplicação de AUI a sistemas colaborativos, mas é rara e está situada na vídeo-conferência [Vertegaal03b]. Nós defendemos que o contexto colaborativo é privilegiado para expandir o corpo de conhecimento sobre AUI e colocamos a hipótese que o desempenho do grupo aumenta se a mediação suportada pelo computador utilizar conceitos das AUI. A nossa motivação está assente nos seguintes postulados:

- Pessoas envolvidas em trabalho colaborativo atendem a muito mais interrupções do que em trabalho individual. Isto está directamente relacionado com a necessidade de manutenção da consciência do grupo e com uma maior variedade de fluxos de informação;
- O trabalho colaborativo é mais fragmentado do que o individual: em vez de executarem uma única e longa tarefa, os membros de grupos tendem a realizar séries de tarefas inter-relacionadas. Esta estratégia melhora a consciência do grupo e permite recuperações mais rápidas em situações de erro;

- No trabalho em grupo as tarefas primária e secundária estão tipicamente relacionadas e podem ambas contribuir para o objectivo comum.

O objectivo deste artigo é reportar a nossa investigação sobre a integração de conceitos de AUI em sistemas colaborativos. Os resultados obtidos encontram-se organizados em duas partes (Secções 2 e 3):

- Estudo dos mecanismos fundamentais de suporte à atenção dos utilizadores de sistemas colaborativos;
- Desenvolvimento de dispositivos especializados na detecção e optimização da atenção do grupo.

Na Secção 4 apresentamos uma ferramenta colaborativa de *brainstorming* electrónico que incorpora os referidos mecanismos de suporte à atenção do grupo. Delineamos ainda uma experiência laboratorial em curso, destinada a testar a hipótese sobre o aumento do desempenho do grupo. Por fim, na Secção 5, discutimos sobre a nossa abordagem e fazemos uma projecção do trabalho futuro.

2. SUPORTE À ATENÇÃO DO GRUPO

O ponto de partida para o estudo da atenção do grupo é o reconhecimento de que os humanos, e, portanto, os *grupos*, têm uma capacidade limitada de processamento de informação. Esta limitação é especialmente relevante em cenários onde os grupos necessitam de atingir um objectivo comum, e ainda mais se forem usados computadores para mediar a comunicação entre os membros do grupo, devido à existência de constrangimentos tecnológicos.

Neste contexto, e tendo por base a análise dos fluxos de informação genericamente disponibilizados pelos sistemas colaborativos, o nosso propósito consiste em identificar e classificar os fluxos fundamentais de suporte à atenção do grupo, nomeadamente os que sustentam as noções de interdependência e de consciência do grupo.

A noção de interdependência é fundamental para compreender como um grupo planeia e estrutura a multiplicidade de actividades realizadas para divergir de ou convergir para um conjunto de conceitos, enquanto a noção de consciência do grupo identifica que estímulos são necessários a uma permanente avaliação do estado do grupo, incluindo as suas capacidades e limitações, motivações, atitudes, coesão, e proximidade do objectivo comum.

A classificação dos mecanismos de suporte à atenção do grupo que propomos fundamenta-se na literatura relacionada e num estudo por nós realizado sobre a optimização dos fluxos de informação suportados por uma aplicação colaborativa [Antunes06]. O resultado desse estudo identifica três tipos de fluxos recorrentes em sistemas colaborativos: *explicit communication*, *feedthrough*, e *back-channel feedback*.

O fluxo *explicit communication* suporta informação produzida por um membro do grupo e intencionalmente dirigida a outros membros [Gutwin02]. Por exemplo, um membro do grupo pode pedir directamente permissão para usar um objecto reservado por outro membro num espaço partilhado. Este fluxo informativo pode ser suportado pelo sistema colaborativo através da distribuição de

informação de um dispositivo de entrada para vários dispositivos de saída.

O fluxo *feedthrough* entrega informação a vários membros do grupo como consequência das acções executadas por um dos membros [Hill03]. Este fluxo é fundamental para criar a consciência do grupo. Por exemplo, um espaço partilhado pode mostrar os menus actualmente seleccionados por cada membro do grupo que está a manipular objectos nesse espaço. Os fluxos de *feedthrough* podem ser suportados através da captura das entradas de cada membro do grupo seguida da distribuição da informação de *feedback* (resposta a um utilizador individual) para os outros membros.

O fluxo *back-channel feedback* suporta informação não intencional enviada por um membro do grupo a outro membro para facilitar a comunicação, indicando em particular que o destinatário está a acompanhar o remetente [Rajan01]. Este fluxo de informação pode ser automaticamente produzido pelo sistema colaborativo, por exemplo utilizando sensores do olhar ou da postura corporal, entre outros [Vertegaal03a, Fogarty05a].

Aparentemente nenhum conteúdo significativo é entregue através do fluxo *back-channel feedback* uma vez que não reflecte cogitação por parte do utilizador. Contudo, como mencionámos antes, do ponto de vista das AUI este fluxo de informação é muito importante pois permite adaptar o sistema aos vários estados de atenção humanos. Por exemplo, numa experiência de vídeo-conferência foram usados sensores de olhar para prolongar a percepção de contacto visual e minimizar as distrações durante o processo de ceder a palavra [Vertegaal03b].

Os sistemas colaborativos podem recorrer a estes diferentes fluxos de informação para manter a atenção do grupo, permitindo interrupções explícitas, mantendo a consciência sobre as actividades em curso, suportando as interdependências, e influenciando as estratégias de trabalho do grupo. Alguns sistemas computadorizados podem combinar todos os tipos de fluxos, em particular os sistemas colaborativos síncronos operados remotamente, enquanto outros podem prescindir da mediação de alguns fluxos, como é o caso dos sistemas de suporte a reuniões face-a-face, que podem dispensar *explicit communication* e *feedthrough*.

3. DISPOSITIVOS DE ATENÇÃO

Iremos agora propor alguns dispositivos para manipular os fluxos de informação em sistemas colaborativos, dado que a quantidade de informação gerada pelos membros do grupo e pelo próprio sistema pode exceder a capacidade de atenção humana e, portanto, fazer diminuir o desempenho do grupo.

O problema que especificamente abordamos é como atenuar a distração decorrente das múltiplas interrupções que caracterizam a colaboração mediada por computador, mantendo simultaneamente a necessária noção de consciência do grupo.

Começamos por categorizar os dispositivos relacionados com a interacção colaborativa em dispositivos de *aware-*

ness e de *coupling*. Os dispositivos de *awareness* são dedicados à detecção e apresentação de informação sobre a actividade colaborativa do grupo. Vários tipos de dispositivos de *awareness*, como vistas de radar e *telepointers*, podem, por exemplo, ser encontrados num estudo sobre sistemas colaborativos organizado em torno da capacidade em transmitir informações sobre quem, o quê, e onde os utilizadores actuam no sistema [Gutwin02]. Nesta categoria de dispositivos também incluímos os sensores que detectam a atenção humana (pelo olhar, postura corporal, e outros) [Vertegaal03a, Fogarty05a], e dispositivos de saída que fazem variar o nível de detalhe de acordo com o foco de atenção do utilizador [Vertegaal03a].

Outra característica dos dispositivos de *awareness* é a de permitirem que os membros do grupo percebam as limitações e o papel mediador da infra-estrutura de comunicação. Isto é particularmente relevante quando a Internet é usada para suportar os fluxos de informação, na qual, por exemplo, os atrasos de *feedthrough* são significativamente maiores e menos previsíveis do que os atrasos de *feedback* [Gutwin04]. Estas limitações também afectam os fluxos *explicit communication* e *back-channel feedback*, e atingem indirectamente as estratégias de colaboração, dado que os atrasos significativos reduzem a consciência do grupo e estimulam as actividades individuais.

Os dispositivos de *coupling* permitem que os membros do grupo façam a separação entre as acções executadas e a informação de consciência do grupo [Dewan95]. Consideramos dois tipos de controlo de *coupling*: por um lado, os membros do grupo podem definir na origem qual e quando deve ser tornada pública a informação sobre as acções que eles próprios realizam; por outro lado, o *coupling* pode ser também controlado no destino, através da especificação de filtros que restringem a consciência do grupo a determinados objectos e acções (ex. mover um *viewport* num espaço de trabalho público).

Os dispositivos de *coupling* requerem a discriminação e controlo manuais da informação de consciência do grupo, quer na origem quer no destino, pelo que penalizam o desempenho individual. Contudo, esta desvantagem é compensada pela flexibilidade que oferecem na gestão da quantidade de informação e do número de interrupções, o que pode contribuir para melhorar a atenção do grupo. Esta contrapartida prepara o palco para a apresentação dos dispositivos especializados para gerir dinamicamente a atenção do grupo.

Nós propomos um conjunto de dispositivos dedicados à recolha da informação recebida dos sensores de *awareness* (entradas) associados a cada membro do grupo, e que automaticamente gerem a informação enviada e apresentada pelos respectivos dispositivos de *awareness* (saídas), em concordância com factores psicológicos conhecidos. O aspecto inovador da nossa abordagem, em comparação com a investigação existente em AUI, está na gestão ser feita ao nível da interacção do grupo, o que significa que os dispositivos têm acesso, não apenas à informação que chega a um utilizador, mas também à informação que é

originária do mesmo ou de outros membros do grupo. Os dispositivos de atenção que propomos são:

Time separator. Entrega a informação a apresentar nos dispositivos de *awareness* de saída após ter decorrido um intervalo de tempo desde a entrega anterior. O propósito deste dispositivo é atenuar os efeitos do *psychological refractory period* (ou *attentional blink*) durante o qual o tempo de resposta a um segundo estímulo é consistentemente mais lento [Eysenck00].

Opportunity seeker. Combina informação de consciência do grupo com o estado de atenção do membro do grupo para determinar o instante de tempo mais adequado para entregar a informação a apresentar no respectivo dispositivo de *awareness* de saída. O objectivo deste dispositivo é descobrir oportunidades que minimizem a distração motivada pelas interrupções.

Change emphasize. Salienta mudanças que ocorreram desde a anterior entrega de informação apresentada nos dispositivos de *awareness* de saída. O propósito deste dispositivo é atenuar a *change blindness* facilitando a detecção visual de mudanças para evitar ter de atender a toda a imagem de consciência do grupo [Simons05].

Activity anticipator. Detecta actividade que pode afectar o desempenho do grupo e entrega informação preliminar para ser apresentada nos dispositivos de *awareness* de saída. O objectivo deste dispositivo é preparar os membros do grupo para estarem atentos para resultados colaborativos iminentes, possibilitando tempos de resposta mais rápidos [LaBerge99].

Estes dispositivos de atenção oferecem suporte explícito para mediar a atenção do grupo num leque alargado de cenários colaborativos. Por exemplo, em sistemas colaborativos assíncronos, o *change emphasize* pode ser usado para salientar diferenças que ocorreram entre dois estados discretos do trabalho em grupo; outro exemplo, desta vez em sistemas síncronos, pode ser a utilização do *opportunity seeker* para evitar interrupções enquanto dois ou mais membros do grupo estiverem activamente a ceder a palavra entre si, numa discussão vídeo, áudio, ou de texto.

4. APLICAÇÃO

Descrevemos agora a aplicação de todos os dispositivos de atenção ao desenvolvimento de uma ferramenta de *brainstorming* electrónico. A descrição está organizada de acordo com a distinção que anteriormente fizemos entre fluxos de informação e dispositivos de atenção.

Sobre os fluxos de informação, as interacções dentro do grupo são feitas exclusivamente através de *feedthrough*. Isto é, a ferramenta de *brainstorming* distribui automaticamente ideias para todos os membros do grupo à medida que vão sendo geradas, sem necessitar de *explicit communication* ou de *back-channel feedback* entre um membro do grupo e outro membro.

Sobre os dispositivos de atenção, a ferramenta de *brainstorming* implementa um dispositivo *time separator* que apresenta novas ideias apenas em intervalos de tempo pré-definidos, um *opportunity seeker* que espera que o

membro do grupo pare de escrever para apresentar novas ideias de outros, um *change emphasize* que realça ideias de outros que foram recentemente recebidas, e um *activity anticipator* que detecta inatividade no grupo e alerta os membros que a sessão irá terminar em breve caso nenhuma nova ideia seja proposta.

Esta ferramenta está actualmente na fase final de desenvolvimento e será posteriormente utilizada em diversas experiências laboratoriais com o objectivo de avaliar o efeito dos dispositivos de atenção em sistemas colaborativos. Os dispositivos de atenção instalados na ferramenta serão utilizados como variáveis independentes no teste à hipótese de melhoria de desempenho do grupo.

O cenário colaborativo experimental é caracterizado por um grupo com cerca de 10 participantes reunidos na mesma sala e a usar a ferramenta de *brainstorming*. Cada sessão dura até 60 minutos, durante os quais a ferramenta mede o número de ideias, o número de ideias de outros que foram recebidas enquanto uma ideia estava a ser escrita, e o tempo de inatividade logo após a recepção de novas ideias de outros membros do grupo. No final, cada participante preenche individualmente um questionário sobre a percepção que teve do seu próprio estado de atenção durante a sessão.

5. DISCUSSÃO E TRABALHO FUTURO

A actual investigação em AUI está quase exclusivamente orientada para actividades realizadas por um só utilizador. Contudo, os utilizadores de sistemas colaborativos também podem beneficiar de soluções computacionais que reconhecem explicitamente as limitações de processamento de informação dos humanos, e, portanto, dos grupos, pelo que defendemos que o corpo de conhecimento existente pode ser estendido para esta área. Com este fim em vista empreendemos o estudo da atenção do grupo, identificando os fluxos de informação típicos dos sistemas colaborativos, e lançámos a hipótese que o desempenho do grupo aumenta se as ferramentas fizerem uso de dispositivos especializados de atenção.

Várias questões permanecem em aberto: irá o desempenho do grupo melhorar significativamente? Que dispositivos de atenção serão mais apropriados para diferentes cenários colaborativos? Poderão ser criados grupos maiores mantendo-se os níveis de atenção e produtividade? Estamos a abordar algumas destas questões com uma ferramenta de *brainstorming* electrónico. O caminho está aberto a muitas mais experiências e aplicações.

6. AGRADECIMENTOS

Este trabalho foi em parte suportado pela Fundação para a Ciência e a Tecnologia, projecto POSI/EIA/57038/2004.

7. REFERÊNCIAS

[Antunes06] P. Antunes, A. Ferreira, e J. A. Pino, "Analyzing shared workspaces design with human-performance models," Aceite para publicação na *CRIWG'06: Proc. of the 12th international workshop on Groupware*, Valladolid, Spain, Setembro 2006.

- [Dewan95] P. Dewan e R. Choudhary, "Coupling the user interfaces of a multiuser program," *ACM Trans. on Computer-Human Interaction* 2(1), 1995, pp. 1–39.
- [Eysenck00] M. W. Eysenck e M. T. Keane, *Cognitive psychology: A student's handbook*, 4^a ed. Psychology Press, 2000, pp. 119–150.
- [Fogarty05a] J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, e J. Yang, "Predicting human interruptibility with sensors," *ACM Trans. on Computer-Human Interaction* 12(1), 2005, pp. 119–146.
- [Fogarty05b] J. Fogarty, A. J. Ko, H. H. Aung, E. Golden, K. P. Tang, e S. E. Hudson, "Examining task engagement in sensor-based statistical models of human interruptibility," *CHI'05: Proc. of the SIGCHI conf. on Human factors in computing systems*, Portland, Oregon, USA, Abril 2005, pp. 331–340.
- [Gutwin04] C. Gutwin, S. Benford, J. Dyck, M. Fraser, I. Vaghi, e C. Greenhalgh, "Revealing delay in collaborative environments," *CHI'04: Proc. of the SIGCHI conf. on Human factors in computing systems*, Vienna, Austria, Abril 2004, pp. 503–510.
- [Gutwin02] C. Gutwin e S. Greenberg, "A descriptive framework of workspace awareness for real-time groupware," *Computer Supported Cooperative Work* 11(3), 2002, pp. 411–446.
- [Hill03] J. Hill e C. Gutwin, "Awareness support in a groupware widget toolkit," *GROUP'03: Proc. of the 2003 international ACM SIGGROUP conf. on Supporting group work*, Sanibel Island, Florida, USA, Novembro 2003, pp. 258–267.
- [Horvitz03] E. Horvitz, C. Kadie, T. Paek, e D. Hovel, "Models of attention in computing and communication: From principles to applications," *Commun. of the ACM* 46(3), 2003, pp. 52–59.
- [LaBerge99] D. LaBerge, "Attention," in *Cognitive Science*, B. M. Bly e D. E. Rumelhart, Eds. San Diego, CA, USA: Academic Press, 1999, pp. 44–98.
- [Rajan01] S. Rajan, S. D. Craig, B. Gholson, N. K. Person, e A. C. Graesser, "AutoTutor: Incorporating back-channel feedback and other human-like conversational behaviors into an intelligent tutoring system," *International Journal of Speech Technology* 4(2), 2001, pp. 117–126.
- [Simons05] D. J. Simons e R. A. Rensink, "Change blindness: Past, present, and future," *Trends in Cognitive Sciences* 9(1), 2005, pp. 16–20.
- [Vertegaal03a] R. Vertegaal, "Attentive user interfaces," *Commun. of the ACM* 46(3), 2003, pp. 30–33.
- [Vertegaal03b] R. Vertegaal, I. Weevers, C. Sohn, e C. Cheung, "GAZE-2: Conveying eye contact in group video conferencing using eye-controlled camera direction," in *CHI'03: Proc. of the SIGCHI conf. on Human factors in computing systems*, Ft. Lauderdale, Florida, USA, Abril 2003, pp. 521–528.

Online Communication in a Multimedia Course

Alexandra Rentróia-Bonito, André Martins, Tiago Guerreiro, Vitor Fernandes, Joaquim A Jorge
Dep. Eng^a. Informática, IST
Av. Rovisco Pais, 1000 Lisboa
{arentroia, apmart, tjvg, vmnf, jaj}@immi.inesc.pt

Abstract

Communication is a critical component of human interaction in order to satisfy basic human needs. This is especially true in technology-supported learning environments, where establishing multiple communicational channels between instructors and students is a requisite to enhance learning. Therefore, specific intervention strategies are required to effectively adjust each channel to satisfy students' communication needs while learning. However, not all online communication channels achieve the goal of including students into their learning communities. Thus, designers should understand what works best in facilitating effective communication and learning. The bottom line lies in how to foster online communication and what impact it may have on effective learning. This paper aims at exploring student use of asynchronous and synchronous communication tools, namely fora and chat tools, and their participation within a technology-supported Multimedia course. We present ongoing research results related to this specific issue, together with implications for designing b-learning experiences, lessons learnt and future work. By this we aim at contributing to understand the impact of online communication tools on student behavior and learning results.

Keywords

B-learning, e-learning systems, HCI, asynchronous and synchronous communication, forum, chat

1. INTRODUCTION

Inclusive learning communities is a desired goal for designers of technology-enhanced learning experiences. Feelings of isolation, high drop-out rates, dissatisfaction, poor learning and low return-on-investment are still likely outcomes that must be addressed to attain equitable, educational, economical and efficiency goals [Mehlen05]. Designing technology-supported learning environments, that are learner-, community-, knowledge- and assessment-centered, is a challenging task because of the multidimensional nature of learning online [Costa05, Ander04]. Monitoring these interactions gives feedback to designers that contributes to the improvement of the people-system fit of the learning experience. This can be done by adjusting the interactivity of content and facilitating proper communication tools to address students' specific needs.

This paper focuses on the students and instructors interactions in a blended-learning experience; specifically, our main objective is to explore the relationship between online participation of students (by posting or viewing messages) and their performance regarding course project. To this end, we analyzed how these variables related during a one-semester Multimedia Production (MPC) course by looking at respective recorded system data; interpret the results identifying key issues and lessons learnt. Our main contribution is to better understand how much specific intervention strategies can contribute to student performance in online learning environments. In

the remainder of this article, we present our conceptual framework, some results gathered and discuss the implications for design. Finally, we present conclusions and ideas for future work.

2. CONCEPTUAL FRAMEWORK

Online communication can be defined as an interchange of ideas and opinions [Webst93] mediated by technology. While communicating involves a sender and a receiver, a communicative act involves relevant content to both parties: one that has the knowledge or information, the other has the desire to learn or be informed. This is required to start "transmitting" messages. In online learning environments, these two roles are easily and rapidly interchanged, which suggests a distributed tutoring operating model between instructors and advanced students in subject matter, be it related to technology or knowledge domain. This is relevant to form and sustain learning communities because of the possibilities to learn interpersonal skills as well as tacit knowledge [Ander04]. This is particularly important, and effective, if e-Learning tools support this bi-directionality and also protect the exchange of information within the community. These specific traits of learning online, if not well managed, drive social isolation, high student anxiety and frustration levels, and drop-out rates [Hambu03]. But, what intervention strategies to use in order to minimize these unwanted effects and benefit from expected results of learning online?

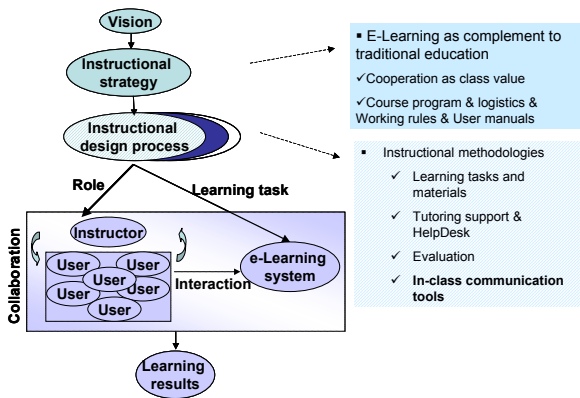


Figure 1 - Conceptual Framework

As can be seen in Figure 1, based on previous works [Rentr03:04:06], the fit between the organizational macro and micro-levels should be articulated in order to create/sustain proper operational context for learning and communicating online. This implies articulating instructional processes, people skills and system functionalities. This can be done through the proper definition of role's behaviors, responsibilities and learning tasks. This way, instructional design process frames expected learning results setting goals in context structuring interactions between learners performing specific roles and defined tasks. Since educating is mostly a matter of communicating, for the remaining part of this paper, we focus on the communicational aspects of the people-system fit within a real learning context, which normally takes place by using fora and chat rooms. By definition, a forum is a medium for public discussion; a public meeting or lecture involving audience discussion [Webst93]. This is an asynchronous way of communicating that supports bi-directionally student-instructor interactions structured around a topic of interest for both parties.

Our framework takes into account instructional strategies and methodologies and their impact on the person-computer interactions and collaboration among them in each defined learning situation. The expected usage of in-class communication tools is the expression of how instructors expect to make users feel part of the learning community and achieve instructional goals. This is because social interaction, both among students and between students and instructors, is strongly related to course satisfaction [Cont04]. In sum, there are challenges for development teams that entail looking at asynchronous and synchronous communications tools and pedagogically adjust their functions to enhance learning. In the next section, we describe our study methodology and results.

3. STUDY METHODOLOGY

The conceptual framework shown in Figure 1 was instantiated into a concrete academic experiment in the form of a one semester **Multimedia Content Production (MCP)** course. Our research group selected and customized an open source Learning Management System [Moodl06] together with a webcast/multimedia archiving tool [epres06], which we called SEMINOLE (*SEaMless INte-*

grated Online Learning Environment). The system was designed to meet three main requirements: learning content management, class webcast and archive, evaluation methods and collaborative work. Its main functionalities were identified based on defined vision, priorities; university's teaching process and analysis of strengths and weaknesses of available LMS (Learning Management System) platforms. While registered students were automatically subscribed to all *fora*, they were able to unsubscribe at will. The system prototype was tested on MCP course last year and improved according to user and technical feedback [Rentr06]. The course was taught during spring semester (2005/06) in two campi of our university to 143 registered students.

The course is divided into practical and theoretical components with several pedagogical activities. In this paper, we present an empirical study on analyzing both synchronous and asynchronous communication methods and the overall results obtained when analyzing interactions in MPC course while students were doing a software group project (three students per group) along the semester with three deliveries. Instructor and teaching assistant lectured in a traditional way while opening different communication channels (online and offline), so that response time to answer questions/doubts concerning class administration or projects was the fastest possible. This response time was between 30 minutes - 12 hours, particularly short in deadline periods. To support the project and students/teachers interactions, SEMINOLE provides, among others: (a) discussion fora where the students can share their concerns about their course administration doubts and project requisites with peers and instructors; (b) a project chat with weekly scheduled tutoring sessions; and (c) a project WIKI where the students can iteratively and collaboratively write their project report. All learning resources/activities contemplate students' and teachers' interactions: the *fora*, as a mean of supporting asynchronous and bi-directionally the discussion of topics, the project chat with its bidirectional synchronously project-related discussion supporting laboratory sessions, and the project WIKI (a wiki is a type of website that allows visitors to easily add, remove, or otherwise edit and change some available content, sometimes without the need for registration. This ease of interaction and operation makes a wiki an effective tool for collaborative authoring) with a mostly unidirectional participation where the instructors only intervene to regularly give feedback about the project progress and report status. The project chat is available while the teacher is present at lab sessions (2 hours / class, 4 classes / week) available for both online and presential students. In addition to these online communication tools, an estimated of two hours/week was spent coaching students on their specific project concerns. All the project chat questions and answers were logged and was available online (like with the *fora*) to be consulted by any student, keeping most of the shared knowledge available to all.

Recorded system data out of 98.7% of registered students was analyzed, excluding the data from drop-out students.

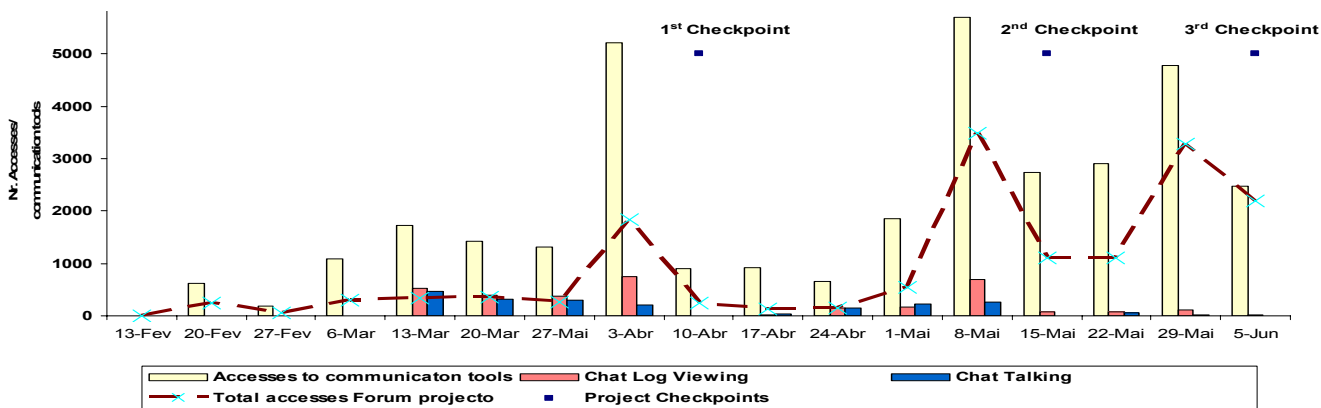


Figure 2 – Project & Thematic Resources Accesses

73% of the students are in their 4th year of a 5-year undergraduate study course. Further, 57% were registered at Campus A; the remaining at Campus B. 79% reported spending more than 2 hours/day using the Internet. 82% reported never having previously participated in a similar b-Learning experience. 58% reported to access Internet at speeds between 512 Kbps – 2 Mbps, 29% at speeds over 2 Mbps and 9% at under 512 Kbps. Almost a quarter of the students held part-time jobs.

4. RESULTS

The assumption underlying this study was that interactions in available communication tools somehow related to student’s participation online. Next, results of students’ participation in each available communication tool in MCP course are presented. All the students participated in the project forum adding 49 new discussions, 308 new posts and viewing 8,783 times existing posts. Regarding the project chat, 90% of the students participated, 36% talking and contributing and 60% viewing available logs. It is worth noticing that, although chat talking decrease as project checkpoints arrived, the fact is that most users preferred asynchronous resources as viewing chat logs and fora. This low-talking rate suggests that most students may not be comfortable expressing themselves online or did not believe it will work well. We can also notice that chat talking peak far from project checkpoint dates as the students use this mean with a more relaxing attitude. The restrictions over synchronous meeting, its demands on time commitments and students’ expectations on their usefulness regarding learning objectives acted as an inhibitor to use chat tool at deadline points. On the contrary, asynchronous tools and instructor’s availability, expressed by a fast response rate (less than 8 hours at stress points, including weekends), worked as a motivation to posting, which increase just before deadlines as depicted in Figure 2. Further studies should explore the usage of whiteboards during lab sessions as a complement of synchronous communication tool to impact learning effectiveness.

At the 6th week of the course, 55% of participating students reported to have posted nothing in existing *fora*. This situation changed in the next month after MCP thematic forum creation (Figure 3). In addition, this forum had an evaluation component performed by peers and

supervised by the instructors. This forum opened three times, starting at the middle of the semester, with a new discussion every three weeks. 67% of the students posted in this forum in response to the three MCP related topics with an average of almost three posts per participating students, fulfilling the defined pedagogical goal. 33% didn’t participate due to overlapping with project checkpoints and other competing course evaluations (about 73% of registered students frequented more than 5 courses on this semester).

We also set up a Q&A forum to clarify students concerns about course management. We had 43 discussions open with an average of three posts per registered student. Also, each student viewed this forum 18 times, on average. Once again instructor’s estimated response time played an important role in students searching for course information and solutions to problems. On the 3rd of April, we created a social chat as suggested by students for them to socialize online. In practice, this chat was just mildly used at the beginning for students tutoring each other during lab sessions. This low usage may be related to students’ concerns for privacy, existing emotional bonds among them, and low perceived usefulness of this communication tool regarding learning objectives. Next steps should explore ways to enhance its usage with class purposes.

Though the sample size is a limitation of this study, reducing the likelihood of discovery significant relationships between analyzed variables; there are two insights to further exploration. First, by looking at a 2 x 2 contingency table using project grades and project forum access frequency, it seems that grades greater or equal than

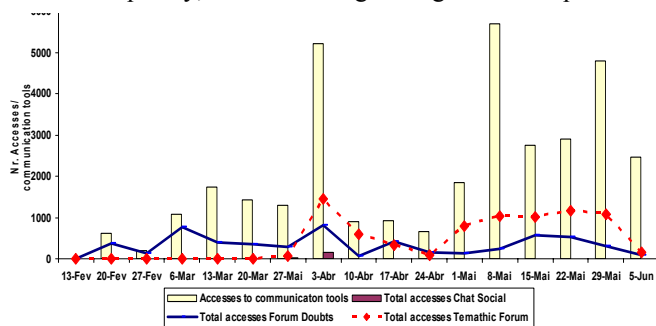


Figure 3 – Accesses to Social and Thematic Resources

15 (out of 20) may be related to accesses greater or equal to 100. Likewise, accesses below 100 may be related to grades below 15 [Montg99]. To test this hypothesis properly, further studies are required with larger samples. Second, the impact of face-to-face tutoring was relevant for learning effectiveness. During the semester, we estimated that a total 200 hours of tutoring was given on two *campi*, covering registered students at different degrees. Some students were inclined to advance in their project work by directly asking to teaching assistant in face-to-face short meetings with low online participation. Again, further studies are needed to analyze the relationship between online participation, learning and presential tutoring.

5. IMPLICATIONS FOR DESIGN

This experience entails four implications for courseware and design as identified. First, from the pedagogical point of view, presential tutoring is still a critical factor when designing technology supported learning experience. Second, availability of instructor and high response rates across all established course *fora* enhances student participation. However, there are still students with low online participation rates that require extra attention from instructors to assure class learning. Third, establishing initial rules and instructions to participate in *fora* and chat rooms helped students to form mental models about functioning and also their expectations regarding timing of response from instructors. Besides, student population grew about five times compared to last year's. This fact demanded a new working method with increased availability supported on better time management and organization skills. Considering this aspect, chat logs and *fora* made it possible for all questions and answers to be kept available as a knowledge base, which was of great value to students as shown by previously mentioned viewing rates. Also, recurring questions were normally redirected to this knowledge base as to peers knowledgeable in the subject's matter, reinforcing the notion of *distributed tutoring*. Last, but not the least, social synchronous communication should be created at the course beginning when students feel the need to meet each other (i.e. Creating project groups).

6. CONCLUSIONS

As previously mentioned, the goal of this work was to present some results from analyzing the interaction between students and instructors by using an online learning environment within a real instructional setting. We conclude that this b-learning experience was efficient and effective in two specific ways. First, a ratio of about 33 students per instructor was possible compare with 9:1 in similar course last year (excluding content production and improvements in course management). Furthermore, drop-out rate was 1.3% among registered students. Also, there was an average of 6 posts made by students per 1 made by instructors. To end, at the timing of this writing, over 90% of the students are expected to make a passing grade judging by similar patterns at a corresponding time last year. However, new issues emerged, as a conse-

quence of this increases in the number of students, such as availability of resources (equipments and facilities), backoffice support to author and timely produce required learning contents in different formats and modes, coordination of support activities at organizational level, helpdesk, adoption of new work methods and a different skill mix in development teams. A second conclusion is that asynchronous communication tools work best to satisfy student needs as they are always available. For this to be true, responses must be timely, short and questions should be fully addressed. This way, well managed asynchronous tools can help build a rich knowledge base resource by capturing what users want to know about course management and subject matter in specific learning situations.

7. REFERENCES

- [Ander04] Anderson, T. Toward a theory of online learning. In *Theory and practice of online learning* (pp. 33-60). Athabasca, AB: Athabasca U. 2004.
- [Costa05] Costabile, M. et al., On the Usability Evaluation of E-learning Applications. IEEE, 2005.
- [Cont04] Contreras-Castillo et al., C. Supporting informal interaction in online courses. E-Education applications: Human factors and innovative approaches. Idea Publishing Group, 2004.
- [Dix98] Dix, A. et al., Human-Computer Interaction. 2nd. edition. UK: Prentice Hall, 1998.
- [epres06] ePresence lecture webcast and recording component. www.epresence.tv (retrieved May 2006).
- [Hambu03] Hamburg, H. et al., Open web-based learning environments and knowledge forums to support disabled people. International Conference KES 2003, (pp. 1208-1215). Oxford, UK, 2003
- [Montg99] Montgomery, D. & Runger, G. Applied statistics and probability for engineers. 2nd edition. John Wiley & Sons, Inc. USA. 1999.
- [Moodl06] Moodle Open Source LMS. Available at www.moodle.org (Retrieved March 2006).
- [Mehlen05] Mehlenbacher, B. et al., Usable E-Learning: A Conceptual Model for Evaluation and Design. NC State University, 2005.
- [Rentr03] Rentroia-Bonito, M. A. & Jorge, J. A. An Integrated Courseware Usability Evaluation Method. (KES 2003). Sept 3-4. Oxford, UK: Lecture Notes in Computer Science, 2774 (2). 208-214.
- [Rentr04] Rentroia-Bonito, M. A. & Jorge, J. (2004a). Towards predictive models for e-learning: What have we learned so far? In E-education applications: Human factors and innovative approaches. Ghaoui, C. (Ed.). USA: Idea Group Publishing, 220-234.
- [Rentr06] Rentroia-Bonito, M. A. et al., Web-based support for resource-effective e-learning. WEBIST'06.
- [Webst93] Webster's new encyclopedic dictionary. Black Dog & Liventhal Publishers Inc. NY. 1993.

Engenharia Reversa de Sistemas Interactivos Desenvolvidos em Java2/Swing

João Carlos Silva^{1,2} José Creissac Campos¹ João Alexandre Saraiva¹

¹ Departamento de Informática/CCTC, Universidade do Minho

² Escola Superior de Tecnologia, Instituto Politécnico do Cávado e do Ave

jcsilva@ipca.pt, {jose.campos, jas}@di.uminho.pt

Resumo

A manutenção e evolução de sistemas interactivos dá origem a problemas importantes que afectam a sua eficiência e eficácia. Neste trabalho pretende-se combinar programação funcional, programação estratégica, code slicing e modelos com semântica formal na tentativa de fortalecer a tese de que a aplicação destas metodologias e tecnologias, em conjunção com um processo de engenharia reversa de código fonte, permite melhorar significativamente o suporte à manutenção de sistemas interactivos.

Palavras-Chave

Interface, Programação Funcional, Code Slicing, Programação Estratégica, Modelos, Haskell, Java/Swing

1. Introdução

Um dos problemas que afectam o sucesso dos projectos de engenharia de software que envolvem uma componente de interacção com o utilizador é a usabilidade [Campos 04]. Vários estudos têm sido realizados na área de métodos formais aplicados à análise da usabilidade [d'Ausbourg 98, Bumbulis 95]. A norma ISO DIS 9241-11 define a usabilidade de um sistema como a eficácia, eficiência e satisfação com que utilizadores atingem determinados objectivos em ambientes específicos. A eficácia diz respeito à possibilidade (ou não) de o utilizador poder atingir os seus objectivos utilizando o sistema num dado contexto. A eficiência tem a ver com o maior ou menor esforço que o utilizador deverá aplicar para atingir esse objectivo. A satisfação é uma medida subjectiva do grau de agradabilidade na utilização do sistema. Avaliar a qualidade de um sistema interactivo implica avaliar a facilidade de utilização do mesmo.

Para que as interfaces tenham uma boa usabilidade é necessário realizar previamente um desenho e uma implementação adequada. Existem ferramentas para auxiliar o desenvolvimento rápido de interfaces através da programação visual das componentes gráficas. Todavia, as interfaces continuam muitas vezes difíceis de entender para os utilizadores finais. Em muitos casos, os utilizadores têm dificuldades em identificar todas as funcionalidades oferecidas pelo sistema, ou dificuldades em alcançar tais funcionalidades.

Tradicionalmente, a área da interacção homem-máquina tem-se preocupado com aspectos relacionados com o desenho de interfaces, enquanto os engenheiros de

software têm principalmente considerados aspectos de implementação. Existe um desencontro entre estas duas áreas, o que leva a que a qualidade do software não seja a melhor. Torna-se necessário que os engenheiros de software tenham em mente preocupações acerca do desenho das interfaces aquando do desenvolvimento de sistemas interactivos.

Hoje em dia é também comum falar-se dos custos na manutenção de sistemas interactivos. Segundo [eES80], 70% do esforço de programação concentra-se na manutenção. Acontecem sempre situações em que existem problemas com software já desenvolvido. Por exemplo, os requisitos iniciais e as tecnologias estão em constante mudança o que implica a alteração do software. Nos primeiros desenvolvimentos de soluções informáticas, a disciplina da engenharia de software concebia o processo de desenvolvimento dos sistemas através de metodologias pouco flexíveis e orientadas fundamentalmente para projectos novos. Rapidamente se observou que os sistemas informáticos têm um período de vida bastante extenso, sendo necessário adaptar os referidos sistemas às diversas evoluções, tanto funcionais como tecnológicas. A engenharia de software tem respondido a estas necessidades adaptando e promovendo metodologias e processos que visam o desenvolvimento de soluções informáticas num processo em espiral [Boehm 88], tentando aumentar a rapidez com que são incorporados novos requisitos (ou alterações de requisitos) nas soluções informáticas.

Nestas situações a utilização de engenharia reversa pode contribuir para a resolução de tais problemas [Memon 03, ES03]. Tendo em vista explorar este aspecto, pretende-se com este trabalho desenvolver uma ferramenta para a

engenharia reversa de sistemas interactivos.

O projecto IVY¹, no qual se enquadra este trabalho, tem como objectivo desenvolver ferramentas para suportar uma abordagem ao desenvolvimento de sistemas interactivos em que se procura facilitar a comunicação entre duas comunidades: a Interação Humano-Computador (IHC) e a Engenharia do Software [Campos 03]. O trabalho já realizado nesta abordagem [Silva 06] é baseado em modelos e pretende possibilitar aos engenheiros de software uma maior autonomia na consideração de aspectos relacionados com o comportamento do sistema.

Neste artigo, apresentamos os resultados do trabalho de investigação realizado na área da engenharia reversa de sistemas interactivos a partir do seu código fonte. Comparativamente com o trabalho já publicado [Silva 06], apresentam-se aqui novos avanços relacionados com a análise comportamental de sistemas interactivos. O protótipo desenvolvido combina várias funcionalidades, nomeadamente: programação estratégica, slicing de programas e modelos abstractos. Pretende-se que a ferramenta seja capaz de contemplar os seguintes pontos:

- gerar modelos do comportamento dos sistemas interactivos a um nível adequado de abstracção;
- Modelar a interacção entre o sistema e o utilizador;
- Ser capaz de gerar modelos para sistemas com elevado grau de dinamismo.

A partir de um qualquer programa JAVA, extraímos dois tipos de modelos. Cada modelo simula o comportamento interactivo induzido por invocações de rotinas da biblioteca gráfica SWING. Esses modelos são designados por interactores e máquinas de estados.

Na secção 2 descrevemos sucintamente o projecto IVY. A seguir na secção 3 descrevemos alguns trabalhos alternativos. Na secção 4 explicamos as técnicas utilizadas no processo de engenharia reversa de interfaces gráficas. A secção 5 descreve duas representações distintas para abstrair o comportamento de sistemas interactivos. A secção 6 apresenta os resultados obtidos com a aplicação do protótipo a um sistema de reduzida dimensão. Finalmente, a secção 7 aponta algumas conclusões assim como algumas orientações para trabalho futuro.

2. O Projecto IVY

O projecto IVY surge na sequência do desenvolvimento da ferramenta I2SMV [Campos 01], a qual permite verificar modelos de sistemas interactivos através do sistema *SMV model checker* [McMillan 93]. O objectivo do projecto consiste no desenvolvimento de uma ferramenta para a análise do comportamento dos sistemas interactivos na fase de desenho. Pretende-se que a ferramenta dê suporte ao processo de modelação e análise através de editores para modelos e propriedades, e visualizadores para a análise dos resultados obtidos no processo de verificação (cf. Figura 1).

¹<http://www.di.uminho.pt/IVY>

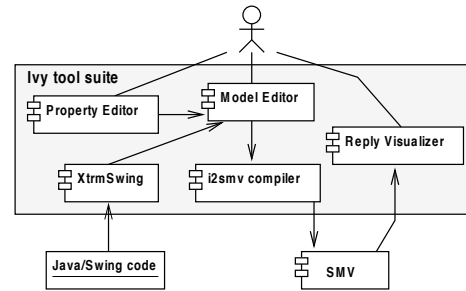


Figura 1. IVY architecture

Através do projecto IVY pretende-se criar uma abstracção na qual os modelos dos sistemas interactivos possam ser mais facilmente desenvolvidos e analisados. Encontramos presentemente a explorar o uso de técnicas de engenharia reversa para permitir a geração de modelos a partir do código fonte. Deste modo, pretende-se analisar as aplicações já existentes bem como dar suporte ao processo de re-engenharia aquando de uma manutenção ou migração. Neste último caso, pretende-se garantir que o novo sistema tem as mesmas características do que o anterior.

3. Trabalhos Alternativos

Nos últimos anos, vários autores investigaram a análise da usabilidade dos sistemas interactivos utilizando modelos e técnicas formais de raciocínio. O uso de engenharia reversa tem sido explorada no sentido de extrair tais modelos directamente a partir dos sistemas interactivos legados. Várias técnicas têm sido apresentadas. Uma abordagem típica consiste em executar o sistema interactivo e automaticamente armazenar os seus estados e acções. Por exemplo, Memon [Memon 03] descreve uma ferramenta que extrai automaticamente dados relacionados com os componentes gráficos, suas propriedades e valores. Chen [Chen 01] propõe uma técnica baseada em especificações para testar interfaces. Os utilizadores manipulam especificações de teste representadas por máquinas de estados finitas as quais são obtidas através da execução do sistema. Por sua vez Systa estuda e analisa o comportamento em tempo de execução de código Java através de um processo de engenharia reversa. A partir da execução do software sobre um *debugger*, Systa consegue gerar um digrama de estados, o qual é utilizado para analisar o comportamento de uma classe, um objecto ou um método [Systa 01].

Uma alternativa que surge no âmbito da engenharia reversa de sistemas interactivos é a análise estática. O processo baseia-se na análise do código de uma aplicação em vez da sua execução, tal como acontece no caso anterior. Por exemplo, d'Ausbourg [d'Ausbourg 96] investigou a análise estática do sistema de janelas X11 [Heller 94]. Nesse trabalho, os modelos salientam os eventos que podem surgir para um dado componente da interface, como por exemplo pressionar uma tecla.

Moore [Moore 96] descreve uma técnica para parcialmente automatizar o processo de engenharia reversa de sistemas interactivos. O resultado deste processo é um modelo para testar a funcionalidade das interfaces gráficas. O trabalho desenvolvido permite com base num conjunto de regras detectar componentes interactivos de código legado. Merlo [Merlo 95] propõe uma abordagem similar. Em ambos os casos é utilizada uma análise estática.

4. Técnica de Engenharia Reversa de Sistemas Interactivos

As contribuições deste trabalho consistem num modelo e numa arquitectura applicacional que suportem a engenharia reversa de sistemas interactivos. A técnica apresentada nesta secção permite construir uma abstracção da interface de um qualquer código *Java* legado. Este processo identifica os dados e acções envolvidas na interface, bem como as interacções entre os vários componentes da interface. Estes componentes incluem objectos e acções da interface. Para construir uma abstracção da interface, a partir da sua implementação em *JAVA*, utilizamos uma combinação de várias técnicas, nomeadamente: programação estratégica, slicing de programas e modelos abstractos. O objectivo é detectar componentes na interface através de estratégias funcionais e modelos com semântica formal.

4.1. O Processo

O protótipo especificado neste trabalho (cf. Figura 2) tem como ponto de partida código fonte *JAVA2/SWING* de sistemas interactivos desenvolvidos através do ambiente integrado de desenvolvimento *NETBEANS*. Tal como referido na secção 1, existem três objectivos essenciais que o protótipo deve satisfazer:

1. Engenharia reversa do código para um nível adequado de abstracção;
2. Modelação de interacção;
3. Modelação de interfaces com elevado grau de dinamismo.

A estrutura deste protótipo é composta por vários passos (cf. Figura 2). Foram utilizadas várias funcionalidades disponibilizadas pela *UMINHOHASKELLLIBRARY*², de entre as quais destacam-se as seguintes:

- Gramática no formato *SDF* da linguagem *JAVA2* e geração de um *parser* para a linguagem;
- Geração automática de um conjunto de tipos de dados abstractos em *HASKELL* [Jones 99] para a representação de código *JAVA2/SWING* segundo uma árvore abstracta de sintaxe;
- Extracção do código *SWING* através de técnicas de slicing construídas utilizando programação estratégica para efectuar travessias da árvore original [Visser 03].

²Conjunto de bibliotecas desenvolvidas na linguagem de programação *HASKELL* pelo Departamento de Informática da Universidade do Minho

A aplicação de algumas funcionalidades da *UMINHOHASKELLLIBRARY* permite deste modo obter uma plataforma funcional para a extracção e manipulação de quaisquer expressões contidas em código *JAVA2/SWING*. Sobre este modelo, foi desenvolvido um protótipo o qual permite inverter código *JAVA2/SWING* para um nível adequado de abstracção.

4.2. Slice de Interfaces

Tal como apontado na secção anterior, para extrair um modelo de uma interface a partir do código é necessário construir uma função de slicing de modo a isolar as instruções relacionadas com a interface. A abordagem habitual consiste em escrever uma função recursiva para executar uma travessia da árvore abstracta de sintaxe (*AST*) de um programa *JAVA*, obtendo como resultado uma sub-árvore com as instruções da interface. Para simplificar o desenvolvimento das funções de travessia, utilizamos uma abordagem diferente, isto é, aplicamos programação estratégica. Neste estilo de programação, existe um conjunto pre-definido de funções genéricas para a travessia de qualquer *AST* incorporando várias estratégias de travessias possíveis (i.e. *top-down*, *left-to-right*, etc). Estas funções permitem ter em conta somente os aspectos de interesse, permitindo assim considerar unicamente as partes relevantes (no nosso caso a linguagem *SWING*).

4.3. A Arquitectura Applicacional

O protótipo foi desenvolvido na linguagem de programação funcional *HASKELL* [Jones 99] e através da plataforma descrita no ponto anterior, permite realizar *slicing* [Tip 95] de qualquer árvore representando código *JAVA2/SWING*, possibilitando assim extrair sub-partes da árvore.

O protótipo extrai e manipula em primeiro lugar o método *INITCOMPONENTS* o qual define todos os componentes gráficos da aplicação.

A seguir, o protótipo extrai do respectivo método toda a informação relacionada com a parte gráfica da aplicação em estudo. Por exemplo objectos das classes *JButton*, *JLabel*, *JTextField*, *JProgressBar*, *JComboBox*, *JSlider*, etc. Para cada tipo de objecto extrai-se somente a informação relevante para este processo de engenharia reversa.

A título de exemplo, descreve-se brevemente como se pode efectuar o slicing em *Strafunski* de sub-expressões *SWING* de um qualquer programa *JAVA*. Para o caso dos objectos definidos através da classe *JButton*, extrai-se da árvore somente as instruções responsáveis por criar esses objectos bem como aquelas que invocam estes métodos:

1. Instâncias da classe *JButton*: As instâncias da classe *JButton* implementam-se via instruções com o seguinte padrão:

```
VarObjName = new java.x.swing.JButton();
```

A partir da árvore, extrai-se então todas as expressões que satisfazem este padrão e guarda-se para cada expressão o nome da variável utilizada.

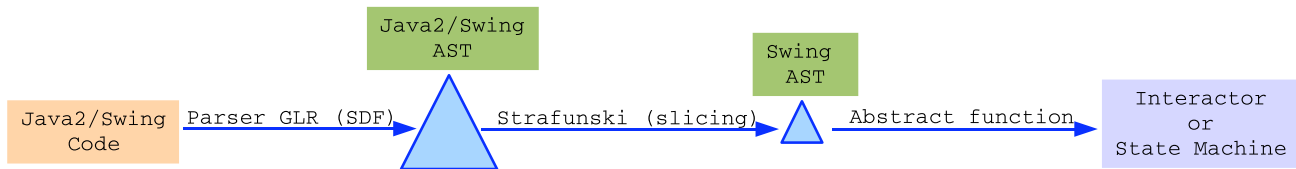


Figura 2. O processo de engenharia reversa

```

getJButtons :: (Term t) => t -> [[Id]]
getJButtons types = jButton
  where
    assignments = getAssig types
    jButton = [a | (a,b) <- assignments,
                 (b==["javax", "swing", "JButton"])]
    
```

2. Método *setText* das instâncias da classe *JButton*: A invocação do método *setText* em objectos definidos com a classe *JButton* implementa-se via instruções com o seguinte padrão:

```
VarObjName.setText(text)
```

A partir da árvore, extrai-se então todas as expressões que satisfazem este padrão e para cada expressão guarda-se o nome da variável utilizada bem como o método invocado e o texto colocado sobre o objecto.

```

getJButtonsSetText :: (Term t) =>
[[Identifier]] -> t -> [[Id],[Id]]
getJButtonsSetText vjid types = jButtonText
  where
    nmi = getNamesMethodsInvocation types
    jButtonText = [(a,["setText"]++(getStr d))
                  | a <- vjid, (c,d) <- nmi,
                    (a++["setText"]) == c]
    
```

3. Método *addActionListener* dos objectos das instâncias da classe *JButton*: A invocação do método *addActionListener* em objectos definidos com a classe *JButton* implementa-se via instruções com o seguinte padrão:

```
VarObjName.addActionListener(expression)
```

A partir da árvore, extrai-se então todas as expressões que satisfazem este padrão.

```

getAddActionListener :: (Term t) =>
[[Id]] -> t -> [[Id],[Id]]
getAddActionListener vjid types = jButtonText
  where
    nmi = getNMI types
    jButtonText = [(a,["addActionListener"]++
                    (concat (map fst (getNMI d))))
                  | a <- vjid, (c,d) <- nmi,
                    (a++["addActionListener"]) == c]
    
```

A seguir determina-se o fecho transitivo de todos os métodos invocados a partir do método *ADD ACTION LISTENER*. Com base no fecho obtido, determina-se assim recursivamente todos os objectos gráficos com os quais o botão está relacionado.

```

getTMB :: (Term t) =>
[(Header,Body)] -> t -> [(Header,Body)]
getTMB t tall =
t++(concat (exp2++exp3))
  where
    nmi = getNMI t
    exp2 = [getMethod a tall | (a,b) <- nmi]
    exp3 = [getTMB [b] tall | b <- exp2]
    
```

5. Relacionando todos os Objectos Gráficos

Após extracção dos dados dos objectos gráficos assim como todos os dados dos métodos sobre estes executados, é então possível gerar representações abstractas das interfaces. Nesse sentido, o protótipo permite criar duas abstracções diferentes: uma máquina de estados e um interactor.

5.1. A máquina de Estados

A máquina de estados permite abstrair o comportamento dos sistemas interactivos. Com esta representação pretende-se abstrair todos os estados gráficos de um sistema interactivo bem como as acções que interligam esses estados. A representação consiste num grafo. Nesse sentido optou-se por utilizar a ferramenta *GRAPHVIZ*³ para a geração do grafo. Em *GRAPHVIZ*, os grafos são elaborados com base numa linguagem própria pelo que foi desenvolvido um módulo auxiliar responsável por produzir o código necessário na linguagem *GRAPHVIZ*.

Toda a informação traduzida para a linguagem *GRAPHVIZ* é obtida com base nos resultados dos *slices* da secção anterior. Assim, cada estado gráfico deduzido da árvore dá origem a um estado gráfico na máquina de estados e cada acção sobre a aplicação dá origem a uma transição entre dois estados. Acrescentam-se também na máquina outros dados específicos do código fonte como por exemplo a inicialização de valores, as condições associadas às acções, etc.

³<http://www.research.att.com/erg/graphviz>

5.2. A Linguagem dos Interactors

Os interactores, tais como desenvolvidos em [Duke 93], são um mecanismo de estruturação de modelos de sistemas interactivos e auxiliam a aplicação de linguagens de especificação de âmbito genérico à modelação desses mesmos sistemas. Os interactores não prescrevem uma linguagem de especificação. Em vez disso, propõem uma estruturação dos modelos que é adequada à modelação de sistemas interactivos e independente da linguagem de especificação utilizada. Utilizando interactores, os modelos são estruturados recorrendo à noção de um objecto que é capaz de apresentar parte do seu estado ao exterior. Assim, cada interactor possui um estado (definido como um conjunto de atributos), um conjunto de eventos a que pode responder ou que pode originar (definido como um conjunto de acções) e uma relação de apresentação que define quais os atributos/acções que são apresentados ao utilizador (marcados com [vis]).

No nosso caso particular, o comportamento dos interactores é definido utilizando uma Lógica Modal de Acções (MAL - Modal Action Logic). Em MAL, temos quatro tipos base de axiomas para definir comportamento:

- axiomas modais permitem definir o efeito das acções no estado do interactor, por exemplo, o axioma $[add_action] \text{consult}' = true \text{ keep}(numero, nota)$ expressa o facto de que, depois da acção *add_action*, o valor do atributo *consult* passa a ser *true*. As plicas são utilizadas para referir o valor de um atributo no estado após a ocorrência da acção (o valor de atributos sem plica é calculado no estado anterior à ocorrência da acção). O operador *keep* é utilizado para indicar que os valores dos atributos passados como parâmetros não mudam.
- axiomas de permissão permitem definir, para cada acção, em que condições ela é permitida, por exemplo, o axioma $per(add) \rightarrow \neg total$ expressa o facto de que a acção *add* pode ocorrer apenas quando o atributo *total* tem o valor falso.
- axiomas de obrigação permitem definir que, em determinadas condições, uma determinada acção tem obrigatoriamente que ocorrer, por exemplo, o axioma $state = open \rightarrow obl(add)$ expressa o facto de que quando o valor do atributo *state* é *open* então a acção *add* tem que ocorrer algures no futuro. Note-se que o axioma não força a acção a ocorrer imediatamente, mas sim que deve ocorrer eventualmente.
- axiomas de inicialização permitem definir o estado inicial do interactor, por exemplo, o axioma $\square add = false, \text{consult} = true$ define os valores dos diferentes atributos no estado inicial do modelo.

O modelo e a arquitectura aqui apresentados definem assim uma abordagem no contexto da engenharia reversa de sistemas interactivos desenvolvidos em JAVA via NETBEANS IDE. A seguir pretende-se descrever a aplicação desta metodologia através de um caso de estudo concreto.

6. Caso de Estudo

A experimentação do modelo e da arquitectura aplicacional sobre um pequeno sistema interactivo desenvolvido com o NETBEANS IDE em Java2/Swing permite validar e demonstrar na prática que o modelo e a arquitectura aplicacional descritos na secção anterior permitem suportar a engenharia reversa de sistemas interactivos desenvolvidos através do ambiente integrado de desenvolvimento NETBEANS.

6.1. Descrição da Aplicação JTURMA

A aplicação em estudo designa-se por JTURMA (cf. Figura 3). Omitimos aqui o código JAVA/SWING e apresentamos apenas a sua representação gráfica.

Esta aplicação permite armazenar e manipular as classificações teóricas e práticas de uma turma. O sistema disponibiliza uma funcionalidade que permite adicionar no sistema os números, os nomes, as classificações teóricas e as classificações práticas de todos os alunos da turma. Para além do armazenamento dos dados, encontram-se disponíveis funções para consultar ou remover os dados de um aluno bem como uma função para terminar a aplicação. No desenvolvimento desta aplicação foram utilizados vários objectos gráficos JAVA2/SWING, nomeadamente objectos das classes JBUTTON, JTEXTFIELD, JLABEL, JPROGRESSBAR, JCOMBOBOX, JSLIDER, JPANEL e GETCONTENTPANE.

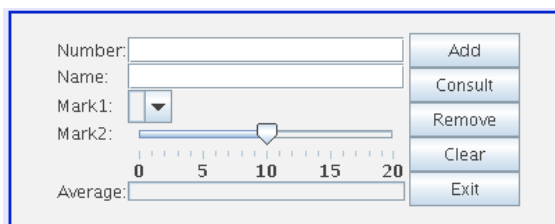


Figura 3. Aplicação JTurma

A execução do protótipo sobre o sistema JTURMA permite gerar de forma automática uma máquina de estados e um interactor. Considerando a Figura 2, quer o interactor quer a máquina de estados correspondem ao resultado do último passo no referido processo de engenharia reversa de sistemas interactivos. A partir da sub-árvore com as expressões SWING e aplicando um conjunto de funções de abstracção, gera-se uma máquina de estados ou um interactor.

6.2. A Máquina de Estados

Na figura 4 apresenta-se parte da máquina de estados gerado pelo protótipo quando executado sobre a aplicação JTURMA.

Nesta máquina de estados, cada estado descreve o conteúdo de uma janela da aplicação num determinado instante. A figura 4 permite-nos visualizar o comportamento da aplicação JTURMA à volta de dois estados

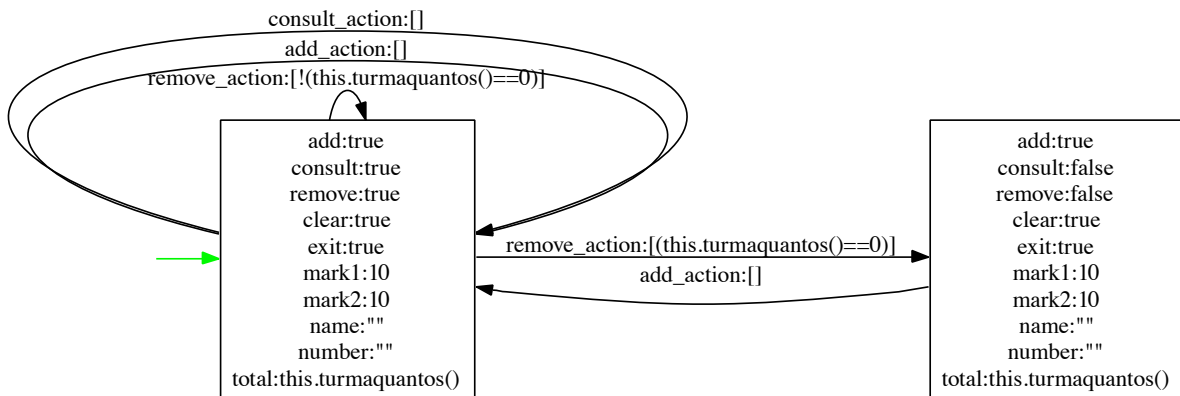


Figura 4. Parte da máquina de estados associada à aplicação *JTurma*

gráficos. Assim a máquina de estados especifica dois estados (conteúdos da janela em dois tempos distintos): à esquerda um estado gráfico com os atributos *consult* e *remove* iguais a *true* e à direita um segundo estado gráfico com os referidos atributos iguais a *false*. Estes dois estados gráficos encontram-se relacionados por transições as quais simbolizam as ações que podem ser executadas sobre um determinado estado. Cada ação encontra-se associada a uma sequência de condições. Estas devem ser satisfeitas para poder executar a respectiva ação. Por exemplo, deduz-se que a remoção de um aluno (a partir do estado esquerdo) implica transitar para o outro estado caso o número de alunos após a remoção seja igual a zero (*this.turmaquantos()==0*), caso contrário a remoção retorna o mesmo estado gráfico.

A partir da máquina de estados pretende-se aprofundar a análise do comportamento de sistemas interactivos calculando automaticamente todas as frases possíveis (conjunto das sequências de transições da máquina de estados) e verificando a seguir se respeitam ou não o modelo de tarefas.

6.3. O Interactor

O protótipo constrói um interactor incluindo o estado inicial da aplicação bem como as suas ações dinâmicas. O interactor contém um conjunto de atributos:

```
interactor JClass
attributes
number,
name: String
mark1,
mark2,
average: Integer
addEnabled,
consultEnabled,
removeEnabled,
clearEnabled,
exitEnabled: Boolean
```

um para cada tipo de componente gráfico, e um por cada um dos estados dos botões. Os nomes dos atributos são

retirados dos nomes das variáveis no código relativas aos componentes gráficos.

O interactor contém também um conjunto de ações:

```
actions
add,
open,
close,
consult,
remove,
clear,
exit,
setText_name(String),
setSelectedItem_mark2(Integer),
setValue_mark1(Integer),
setValue_average(Integer),
setText_number(Integer)
```

uma por cada botão, e uma por cada componente gráfico de *input*.

E finalmente um conjunto de axiomas:

```
[ ]
number="" & name="" &
mark1=10 & mark2=10 & average=0
[ ]
addEnabled=true & clearEnabled=true &
exitEnabled=true & consultEnabled=false &
removeEnabled=false & number="" &
name="" & mark1=10 & mark2=10 & average=0
[add]
number'=number & name'=name & mark1'=mark1 &
mark2'=mark2 & average'=average &
consultEnabled'=true & removeEnabled'=true &
addEnabled'=addEnabled &
clearEnabled'=clearEnabled &
exitEnabled'=exitEnabled
[consult]
number'=number & name'=?ref1? &
mark1'=?ref2? &
mark2'=?ref3? & average'=?ref4? &
addEnabled'=addEnabled &
consultEnabled'=consultEnabled &
```

```

removeEnabled'=removeEnabled &
clearEnabled'=clearEnabled &
exitEnabled'=exitEnabled
[remove]
number'=number name'=name &
mark1'=mark1 & mark2'=mark2 &
average'=average &
addEnabled'=addEnabled &
clearEnabled'=clearEnabled &
exitEnabled'=exitEnabled
[clear]
number'?=?ref5? & name'?=?ref6? &
mark1'?=?ref7? & mark2'?=?ref8? &
average'?=?ref9? &
addEnabled'=addEnabled &
consultEnabled'=consultEnabled &
removeEnabled'=removeEnabled &
clearEnabled'=clearEnabled &
exitEnabled'=exitEnabled
[setText_name(a)]
name'=a & number'=number &
mark1'=mark1 & mark2'=mark2 &
average'=average &
consultEnabled'=consultEnabled &
removeEnabled'=removeEnabled &
addEnabled'=addEnabled &
clearEnabled'=clearEnabled &
exitEnabled'=exitEnabled
...

```

Os dois primeiros axiomas definem o estado inicial do sistema. Os quatro seguintes definem o efeito dos botões na interface. As expressões `?refX?` representam valores que devem ser preenchidos através do editor IVY. Para apoiar o preenchimento do modelo, cada expressão é um apontador para o código JAVA que define o valor a atribuir. O último axioma define o efeito das acções de *input* na caixa de texto *name*.

Ainda que incompleto, este interactor inclui dados relevantes no que diz respeito ao comportamento do sistema JTURMA. A título de exemplo, o quarto axioma, exprime o estado interactivo após executar a acção *consult*. Podemos constatar que os atributos *number*, *addEnabled*, *consultEnabled*, *removeEnabled*, *clearEnabled*, *exitEnabled* mantêm-se inalterados. Por outro lado, os atributos *name*, *mark1*, *mark2* e *average* recebem novos dados.

Após instanciar totalmente o modelo, este poderá ser usado pela ferramenta IVY para verificação do seu comportamento.

Através dos resultados obtidos, demonstra-se que a abordagem descrita na secção anterior, ainda que actualmente com muitas limitações, suporta a engenharia reversa de sistemas interactivos desenvolvidos na linguagem JAVA/SWING.

7. Conclusão

Neste artigo, foi descrita a forma como a programação estratégica e técnicas de *slicing* podem ser aplicados à engenharia reversa de interfaces a partir do código das aplicações. O resultados deste trabalho evidenciam a pos-

sibilidade de inverter interfaces. Um protótipo foi desenvolvido permitindo extrair automaticamente o comportamento de uma aplicação a partir do seu código fonte.

Actualmente a ferramenta permite extrair um sub-conjunto de componentes gráficos e de acções, a partir do qual gera uma máquina de estados e um interactor. Estes modelos permitem então raciocinar acerca da usabilidade e qualidade da implementação. Neste momento, o protótipo considera somente um conjunto reduzido dos componentes SWING. Como trabalho futuro, pretende-se estender a implementação de modo a contemplar interfaces mais complexos.

8. Agradecimentos

Este trabalho é parcialmente suportado pela FCT (Portugal) e FEDER (União Europeia) através do contrato POSC/EIA/56646/2004.

Referências

- [Boehm 88] B. Boehm. *A Spiral Model for Software Development and Enhancement*, volume 21, páginas 61–72. Computer edição, Maio 1988.
- [Bumbulis 95] Peter Bumbulis e P.S.C. Alencar. A framework for prototyping and mechanically verifying a class of user interfaces. *IEEE*, 1995.
- [Campos 01] José C. Campos e Michael D. Harrison. Model checking interactor specifications. *Automated Software Engineering*, August 2001.
- [Campos 03] J. C. Campos e M. D. Harrison. From HCI to Software Engineering and back. Em Rick Kazman, Len Bass, e Jan Bosch, editores, *Bridging the Gaps Between Software Engineering and Human-Computer Interaction, ICSE '2003 workshop*, páginas 49–56, Portland, Oregon, USA, May 2003. IFIP.
- [Campos 04] José C. Campos. The modelling gap between software engineering and human-computer interaction. Em Rick Kazman, Len Bass, e Bonnie John, editores, *ICSE 2004 Workshop: Bridging the Gaps II*, páginas 54–61. The IEE, 2004.
- [Chen 01] J. Chen e S. Subramaniam. A gui environment for testing gui-based applications in java. *Proceedings of the 34th Hawaii International Conferences on System Sciences*, january 2001.
- [d'Ausbourg 96] Bruno d'Ausbourg, Guy Durrieu, e Pierre Roché. Deriving a formal model of an interactive system from its UIL description in order to verify and to test its

- behaviour. Em *Design, Specification and Verification of Interactive Systems '96*, páginas 105–122. 1996.
- [dAusbourg 98] Bruno dAusbourg, Christel Seguin, e Pierre Rochk Guy Durrieu. Helping the automated validation process of user interfaces systems. *IEEE*, 1998.
- [Duke 93] D. J. Duke e M. D. Harrison. Abstract interaction objects. *Computer Graphics Forum*, 1993.
- [eES80] B. Lientz e E. Swanson. *Software Maintenance Management*. Addison-wesley edição, 1980.
- [ES03] P. Iglinski E. Stroulia, M. El-ramly e P. Sorenson. User interface reverse engineering in support of interface migration to the web. *Automated Software Engineering*, 2003.
- [Heller 94] Dan Heller e Paula M. Ferguson. *Motif Programming Manual*, volume 6A de *X Window System Seris*. O'Reilly & Associates, Inc., second edição, 1994.
- [Jones 99] Simon Peyton Jones, John Hughes, Lennart Augustsson, et al. Report on the Programming Language Haskell 98. Relatório técnico, Fevereiro 1999.
- [McMillan 93] Kenneth L. McMillan. Symbolic model checking. *Kluwer Academic Publishers*, 1993.
- [Memon 03] Atif Memon, Ishan Banerjee, e Adithya Nagarajan. Gui ripping: Reverse engineering of graphical user interface for testing. Relatório técnico, 2003. Department of Computer Science and Fraunhofer Center for Experimental Software Engineering, University of Maryland, USA.
- [Merlo 95] Merlo, Gagne, Girard, Kontogiannis, Hendren, Panangaden, e De Mori. Reverse engineering and reengineering of user interfaces. *IEEE Software*, 12(1), 64-73, 1995.
- [Moore 96] M. M. Moore. Rule-based detection for reverse engineering user interfaces. *Proceedings of the Third Working Conference on Reverse Engineering*, pages 42-8, Monterey, CA, november 1996.
- [Silva 06] João Carlos Silva, José Creissac Campos, e João Saraiva. Combining formal methods and funcional strategies regarding the reverse engineering of interactive applications. *DSVIS-2006, Design, Verification and Validation of Interactive Systems*, July 2006.
- [Systa 01] T. Systa. Dynamic reverse engineering of java software. Relatório técnico, 2001. University of Tampere, Finland.
- [Tip 95] Frank Tip. A survey of program slicing techniques. *Journal of Programming Languages*, september 1995.
- [Visser 03] Joost Visser. *Generic Traversal over Typed Source Code Representations*. Tese de Doutorado, University of Amsterdam, Fevereiro 2003.

Representación abstracta de interfaces de usuario DGAUI

Javier Rodeiro Iglesias

Susana Gómez Carnero

Escuela Superior de Ingeniería Informática de la Universidad de Vigo
Campus As Lagoas S/N, Ourense
{jrodeiro,susanagomez}@uvigo.es

Sumario

Las interfaces de usuario tienen un papel crucial en el éxito de una aplicación, a la vez que su desarrollo conlleva un importante coste tanto desde el punto de vista económico como temporal, con estos antecedentes es inevitable pensar que su optimización para conseguir una aceptabilidad máxima y un diseño efectivo sería de gran interés. Para que una interfaz se considere aceptable debe ser agradable a la vista del usuario, hacer lo que tiene que hacer y tener un uso fácil por parte del usuario. En este paper se presenta un modelo de representación de la interfaz que permita establecer validaciones sobre las tres premisas de aceptabilidad de interfaces de modo semiautomático, minimizando los costes de desarrollo y evaluación existentes actualmente. La representación que se propone en el modelo, considera que la interfaz de usuario visual no es una estructura continua sino que está compuesta por elementos finitos discretos, plantea la interfaz como una composición de elementos individuales denominados componentes, de estos componentes se indicará su descripción, las relaciones jerárquicas que se establecen entre ellos y la descripción de los cambios que provoca la realización de un evento sobre un componente dado.

Palabras clave

Diseño interfaces, usabilidad, modelos, representación abstracta

1. INTRODUCCIÓN

Dada la importancia de las interfaces de usuario en el éxito o fracaso de una aplicación y el tiempo y dinero que se invierten en su desarrollo, recordemos que Myers [Myers92] a través de una encuesta a desarrolladores lo cifra en un 48% y Gartner Group [Gartner94] dice que el 70% del esfuerzo de desarrollo de aplicaciones interactivas se dedica a la interfaz, es inevitable pensar que sería interesante poder optimizar las características de las mismas para conseguir una aceptabilidad máxima a través de un diseño más efectivo y consiguiendo de este modo un mejor aprovechamiento de los recursos.

La aceptabilidad de una interfaz de usuario está basada principalmente en tres cuestiones:

1. ¿Es la interfaz agradable a la vista del usuario? En este caso se habla de aceptabilidad de la interfaz basada en la estética de la misma. No se establecen criterios de utilización de la interfaz, si la aplicación subyacente de la interfaz es correcta o carga de memoria del usuario a la hora de utilizarla. Lo fundamental es que el usuario se sienta en un entorno agradable y amigable en cuanto a lo que ve, y que esta interfaz no le sea de ninguna forma repulsiva.

2. ¿Hace la interfaz lo que tiene que hacer? Se determina aquí si la interfaz es útil para el propósito con el cual fue creada, tanto en términos de que haga aquello que el usuario le indica que haga como en términos de que el usuario sea capaz de hacer aquello que desea en la

interfaz. Este criterio está relacionado con las tareas que el usuario tiene que realizar sobre la interfaz.

3. ¿Es fácil su uso por parte del usuario? Aquí se introduce el término de usabilidad [Shackel86] [Eason88] [ISO92] [ISO93] [Nielsen93] [IBM93] de la interfaz. Si un usuario encuentra demasiado complejo el manejo de una interfaz, por muy agradable que esta sea y por muy eficiente que sea el proceso que desarrolla, no la usará. La evaluación del cumplimiento de estos criterios es muy compleja. Esta complejidad viene dada porque el estudio de los mismos suele ser en la mayoría de los casos mediante inspección subjetiva, principalmente basada en opiniones de expertos o en encuestas sobre usuarios [Molich90] [Preece94] [Nielsen94] [Wharton94].

Existe una gran cantidad de técnicas de representación de interfaces de usuario, como indica Vanderdonckt [Vanderdonckt94] muchas de estas técnicas son exportadas de otras áreas como puede ser el diseño gráfico. Otras de estas representaciones vienen de formalismo matemáticos como el álgebra. Debemos examinar estas representaciones para comprobar si pueden ser aplicadas a interfaces complejas y establecer de una forma objetiva el cumplimiento de los criterios identificados anteriormente.

Por ello se plantea como objetivo del trabajo encontrar un modelo de representación de la interfaz que permita establecer validaciones sobre las tres premisas de aceptabilidad de interfaces de modo semiautomático, minimizando los costes de desarrollo y evaluación

existentes actualmente. Para ello en la sección 2 se presenta una alternativa que permite la representación de los aspectos visuales y de comportamiento de la interfaz. En la sección 3 se presenta un ejemplo de aplicación de la notación y en la sección 4 se extraen conclusiones y se plantean los trabajos futuros.

2. REPRESENTACIÓN DGAUI

Se ha realizado una revisión de la literatura existente sobre modelos de representación de interfaces de usuario, centrándose en aquellos modelos que proponían una representación visual de la interfaz y que permitían la definición del comportamiento de la misma [Gomez05], pero considerando los problemas que se han encontrado en dichas representaciones se presenta una solución alternativa.

La representación que se propone, DGAUI, considera que la interfaz de usuario visual no es una estructura continua sino que está compuesta por elementos finitos discretos, plantea la interfaz como una composición de elementos individuales denominados componentes, dichos componentes siguen una estructura topológica jerárquica, de forma que pueden estar contenidos unos dentro de otros.

Para la definición de la IU el diseñador gráfico dispone de un sistema de representación en el que puede:

- Realizar la definición de los componentes de la IU, posibilitando la creación de jerarquías de composición y la definición de nuevos componentes.
- Indicar la composición de componentes individuales para formar la interfaz de usuario concreta con la que el usuario interactúa.
- Representar el diálogo entre componentes dentro de la interfaz de usuario, indicando los eventos a los que atiende cada componente y de qué manera responde la IU a los mismos

Se entenderá que un **componente** de la interfaz de usuario es un elemento individual de la misma que es perceptible por el usuario a nivel de apariencia visual, responde a una acción del usuario, o bien sirve de elemento agrupador de otros componentes.

Cada componente tiene unas características propias que son independientes de la interfaz. Se consideran inherentes al componente, además de poseer unas propiedades relativas a su posicionamiento y comportamiento que se derivan de su participación o papel dentro de la IU en la que están incluidos.

Para una mejor interpretación y procesamiento de la representación de la interfaz parece adecuado estructurarla y las características de XML se adecuan perfectamente.

Por ello se plantea una estructuración de la representación basada en XML mediante la creación de una DTD que permita un parseado sencillo de la estructura de la representación.

Atendiendo a la semántica de la representación, en donde se encuentra por un lado la representación inicial de la

interfaz, y después de un procesamiento, la estructura de los estados de la misma y sus transiciones, parece adecuado el separar asimismo estas dos representaciones. La primera (a la que llamaremos DGAUI-DEF) consiste en una definición detallada de cada uno de los elementos que conforman una interfaz y que es adecuado tener separados para permitir reutilización de la representación. La segunda (a la que llamaremos DGAUI-INT) es dependiente de la primera pues se calcula de la misma, aunque también sería factible construirla directamente si el concepto de diseño de la interfaz se orienta a estados y no a componentes como es el caso de DGAUI.

De esta forma, además, se permite una separación base entre presentación de la interfaz y funcionalidad de la misma. La presentación de la interfaz se haya en la definición de la representación mientras que la funcionalidad se haya en los estados y el tránsito entre estados a través de las acciones de los usuarios sobre los componentes y de los eventos del sistema.

Por tanto, la descripción inicial de la interfaz de usuario, DGAUI_DEF, se realizará mediante un documento XML que contiene la siguiente información:

- Composición de los componentes contenidos dentro de los contenedores.
- Información sobre la representación visual de los componentes.
- Información sobre los eventos y respuestas sobre componentes ante la ocurrencia de ese evento

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT DGAUIDEF (Composicion, Descripcion, Dialogo)>
<!ELEMENT Composicion (Componente+)>
<!ELEMENT Descripcion (Grafico* | Texto* | Enumeracion*)*>
<!ELEMENT Dialogo (ItemDialogo*)>
```

Cada **componente** contenedor está formado por:

```
<!ELEMENT Componente (Alineado?, Equiespaciado?, Subcomponentes, Infi?, Info?)>
```

Indica el nombre del componente, opciones de detalle sobre la alineación de los componentes incluidos, la lista de componentes incluidos y si el componente permite información de entrada o salida.

- *Infi*: El componente realiza introducción de información por parte del usuario. Se indica la información a través de un Id. Es opcional.

```
<!ELEMENT Infi (#PCDATA)>
```

- *Info*: El componente muestra información desde la aplicación. Se identifica la información a través de un Id. Es opcional.

```
<!ELEMENT Info (#PCDATA)>
```


La **descripción** de cada componente de la interfaz indica que puede ser un gráfico, un texto o un gráfico definido por enumeración.

```
<!ELEMENT Descripcion (Grafico* | Texto* | Enumeracion*)*>
```

Componente gráfico: viene definido por su nombre, una primitiva, el estilo, ancho y color de línea, su color de relleno, su posición y su tamaño.

```
<!ELEMENT Grafico ((Rectangulo | Linea | Circulo | Elipse | Poligono)?, EstiloLinea?, AnchoLinea?, ColorLinea?, ColorRelleno?, Posicion?, Tamano?, Datos?)>
```

Tanto los componentes gráficos, como los componentes tipo texto y los gráficos por enumeración tienen un conjunto de propiedades comunes que tendrán valor verdadero o falso. Estas propiedades son:

- Activo: indica si el componente puede responder a eventos.
- Visible: indica si el componente es visible para el usuario en el dispositivo de visualización.
- InfI: permite activar la función de entrada de datos asociada al componente. Es opcional.
- InfO: permite activar la función de salida de datos asociada al componente. Es opcional.

```
<!ATTLIST Grafico
  Nombre CDATA #REQUIRED
  Visible (t | f) #REQUIRED
  Activo (t | f) #REQUIRED
  InfI (t | f) #IMPLIED
  InfO (t | f) #IMPLIED >
```

La primitiva que define el componente puede ser cualquiera de las primitivas de dibujo usadas generalmente en los paquetes gráficos.

La *posición* de un componente determina su posición dentro de su componente contenedor o bien dentro del dispositivo. Dicha posición puede ser fija o relativa indicada mediante restricciones topológicas que se indican en el campo OpCRel, en cualquier caso esta posición viene dada por una coordenada que indica su posición superior izquierda.

```
<!ELEMENT Posicion (Fija | Relativa)>
  <!ELEMENT Fija (Coordenada)>
  <!ELEMENT Relativa (OpCRel?, Coordenada)>
  <!ELEMENT OpCRel (Centrado | (Justificado, Justificado?))>
```

- *Componente de texto:* viene definido por el texto que representa, un tipo, tamaño y color de fuente; además también posee las características de posición y tamaño así como las características de visible, activo, InfI e InfO ya explicadas en componentes gráficos.

```
<!ELEMENT Texto (Txt, Fuente, TamanoFuente, ColorFuente, EstiloFuente, Posicion, Tamano)>
```

- *Componentes gráficos por enumeración:* incluye la ruta al fichero que contiene la imagen correspondiente al gráfico; además incluye valores para indicar el nombre, posición, tamaño, su visibilidad, y actividad y posibilidad de entrada y salida de datos por parte del usuario ya descrito en componentes gráficos.

```
<!ELEMENT Enumeracion (Fichero, Posicion, Tamano)>
```

El **diálogo** sobre los componentes de la interfaz está formado por el componente sobre el que se produce el diálogo, el evento que lo provoca y los cambios efectuados en las propiedades de otros componentes (La respuesta al evento).

```
<!ELEMENT Dialogo (ItemDialogo*)>
<!ELEMENT ItemDialogo (Precondiciones?, Respuesta)>
<!ATTLIST ItemDialogo
  Elemento CDATA #IMPLIED
  Evento CDATA #REQUIRED >
<!ELEMENT Precondiciones (Precondicion+)>
<!ELEMENT Precondicion (#PCDATA)>
<!ATTLIST Precondicion
  Visible (t | f) #IMPLIED
  Activo (t | f) #IMPLIED
  InfI (t | f) #IMPLIED
  InfO (t | f) #IMPLIED >
<!ELEMENT Respuesta (Cambio)+>
<!ELEMENT Cambio (#PCDATA)>
<!ATTLIST Cambio
  Visible (t | f) #IMPLIED
  Activo (t | f) #IMPLIED
  InfI (t | f) #IMPLIED
  InfO (t | f) #IMPLIED
  ProcID CDATA #IMPLIED >
```

La representación de los componentes de la interfaz es fija para la representación de la interfaz. Los aspectos dinámicos producidos por la interacción del usuario con la interfaz están reflejados en la representación del diálogo de los componentes.

En la representación del diálogo se asocia a cada componente el evento al cual responde junto con los cambios que este provoca en la interfaz.

3. Experimentación

Como ejemplo simple se presenta la modelización de un botón Next cuya apariencia sería:

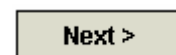


Figura 1: Botón Next

La acción que se podría realizar sobre él sería la selección del mismo, obteniendo la siguiente apariencia:



Figura 2: Botón Next seleccionado

Se presenta a continuación la representación XML del botón.

En la composición se identifican dos componentes independiente Button_Off y Button_On.

```
<DGAUIDEF>
<Composicion>
<Componente Nombre="Button_Off">
<Subcomponentes>
<Cont>Button_off_text</Cont>
</Subcomponentes>
</Componente>
<Componente Nombre="Button_On">
<Subcomponentes>
<Cont>Button_on_text</Cont>
</Subcomponentes>
</Componente>
</Composicion>
```

La descripción presenta los componentes contenedores independientes (Button_Off y Button_On) como dos primitivas rectángulo y los subcomponentes (Button_off_text y Button_on_text) como componentes de tipo Texto.

El fichero de especificación completo de este ejemplo se puede obtener en [Gomez05]

4. Conclusiones y Trabajos futuros

Se ha mostrado una nueva representación que hace especial énfasis en la parte visual de la interfaz, relacionándose muy fuertemente esta parte visual con el comportamiento de la interfaz frente a las acciones del usuario.

Se ha planteado un ejemplo concreto del uso de la notación para un componente simple, pudiéndose deducir de los resultados obtenidos su aplicabilidad sobre múltiples componentes. Siendo factible obtener de forma automática el comportamiento de los componentes.

Como trabajo futuro se plantea la presentación de la notación que permita representar la funcionalidad de las interfaces (DGAUL_INT) así como la construcción de una interfaz compleja como podría ser la de un procesador de texto.

Una vez modelada la interfaz se plantea la extracción del conjunto de tareas de usuario desde la especificación como base para la evaluación de la interfaz.

5. AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto TIN2005-08863-C03-02 y por el proyecto 05VI-C02.

6. REFERENCIAS

- [Eason88] Eason, K. Information Technology and Organizational Change. Taylor and Francis, London. 1988.
- [Gartner94] Gartner Group Annual Symposium on the Future of Information Technology, Cannes 7-10 november 1994.
- [Gomez05] Gómez Carnero, S, Rodeiro Iglesias, J. Aplicación de los sistemas de representación para la sistematización de la validación de interfaces de usuario. *Technical Report TR-LSI-GIG-05-2. Computer Science Department. University of Vigo. 2005.*
www.ei.uvigo.es/~susanagomez/hci.html
- [IBM93] IBM, IBM Dictionary of Computing. McGraw-Hill 1993.
- [ISO92] ISO. Software product evaluation quality characteristics and guidelines for their use.
- [ISO93] ISO. Ergonomics Requirements for Office Work with Visual Displays Terminals: Guidance and Usability.
- [Molich90] Molich R. y Nielsen J. Heuristic evaluation of user interfaces in *Proceedings of ACM CHI 1990*. Seattle, WA, Abril 1990, Pág. 249-256, 1990.
- [Myers92] Myers B. A. y Rosson M. B. Survey on user interface programming, in *CHI'92 Conference Proceedings on Human Factors in Computing Systems* (Bauersfeld P., Bennett J. y LYNCH G. eds.), pág. 195-202. ACM Press, Nueva York, NY, 1992.
- [Nielsen93] Nielsen, J. Usability Engineering. Academic Press, London. 1993.
- [Nielsen94] Nielsen J. y Mack R. L. eds. Usability Inspection Methods. John Wiley and Sons, New York, NY, 1994.
- [Preece94] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. And Carey Tom. Human-Computer Interaction. Addison-Wesley Publishing, Reading, Mass. 1994.
- [Rodeiro01] Rodeiro Iglesias, J. Representación y análisis de la componente visual de la interfaz de usuarios. PhD Thesis. Universidad de Vigo. Septiembre 2001.
- [Shackel86] Shackel, B. Ergonomics in designing for usability. In *M. D. Harrison and A. Monk, editors. People and Computers: Designing for Usability*. Cambridge University Press, 1986.
- [Vanderdonck94] Jean Vanderdonck, Xavier Grillo. Visual techniques for traditional and multimedia layout. June 1994.
- [Wharton94] Wharton C. ET AL. The cognitive walkthrough method: a practitioner's guide en Usability Inspection Methods (NIELSEN J. y MACK R. L. eds.). John Wiley & Sons, New York, NY, Pág. 105-140, 1994.

UML no Desenvolvimento de Sistemas Interactivos

José Creissac Campos António Nestor Ribeiro
Departamento de Informática / CCTC, Universidade do Minho
Campus de Gualtar, 4710-057 Braga, Portugal
{jose.campos, anr}@di.uminho.pt

Resumo

Os processos típicos de análise e modelação de sistemas numa perspectiva de engenharia de software atribuem pouca importância à modelação da camada interactiva. O Unified Process e a linguagem que o suporta (UML) não são excepção. Este artigo propõe uma abordagem, que procura integrar a modelação da camada interactiva no processo de modelação típico baseado em UML, explorando as potencialidades fornecidas pela linguagem.

Palavras-Chave

Desenho baseado em modelos, UML, desenho centrado no utilizador, Unified Process.

1. INTRODUÇÃO

A *Unified Modelling Language* (UML) [Group 05] constituiu-se como a norma para o desenvolvimento de software orientado a objectos. A orientação a objectos é uma abordagem atractiva no que se refere ao desenvolvimento de sistemas interactivos, uma vez que este tipo de abordagem potencia a separação de camadas proposta em modelos como o MVC [Krasner 88]. No entanto, tanto a UML, enquanto linguagem de modelação, como o Unified Process [Jacobson 99], enquanto processo de desenvolvimento, são reconhecidamente deficientes no suporte ao desenvolvimento de sistemas interactivos [John 03]. Face às novidades introduzidas na versão 2.0 da UML torna-se interessante explorar qual a melhor forma de retirar o máximo benefício da linguagem no contexto do desenvolvimento de um sistema interactivo.

Neste artigo apresentamos uma proposta para a utilização da UML no contexto referido. Não propomos a inclusão de novos elementos na linguagem, visto que a nossa motivação se centra em utilizar a norma e procurar as melhores práticas que permitam a sua correcta utilização no desenho da camada interactiva. Este trabalho tem a sua génese na leccionação da linguagem a alunos de licenciatura e foi, em parte, motivado pelas dificuldades sentidas pelos mesmos na articulação entre camada interactiva e camada de lógica de negócio. Assim, exploramos até que ponto a UML pode ser utilizada na resolução dos problemas que um engenheiro de software enfrenta, quais os seus pontos fortes e fracos, bem como os compromissos que se impõem.

2 UML E SISTEMAS INTERACTIVOS

Tal como foi referido, têm surgido diversas propostas de extensão da UML para melhor a adaptar ao desenvolvimento de sistemas interactivos. Algumas destas propostas

vão no sentido de adicionar novas linguagens ao conjunto disponibilizado pela UML [Patern? 00]. Outras no sentido de propôr perfis para a modelação de aspectos dos sistemas interactivos em UML [Nunes 01, Silva 02].

A adopção destas propostas por parte da comunidade tem-se mostrado lenta. Este facto dever-se-á, por um lado à natural resistência a aprender novas linguagens de modelação, e por outro, à resistência a apreender novos perfis face ao potencial de confusão que eles encerram (por se tratar de novas extensões à notação).

Acresce que em alguns casos os perfis se centram na modelação dos objectos gráficos presentes na interface quando esse aspecto começa a estar razoavelmente resolvido pelos IDEs correntes com a possibilidade de edição gráfica das mesmas. Na verdade, a questão coloca-se, não tanto na edição do aspecto gráfico da interface, mas mais na estruturação do diálogo entre utilizadores e aplicações, bem como na estruturação do código que o suporta.

Como a UML se encontra bem estabelecida e é utilizada e suportada por uma vasta comunidade de utilizadores, torna-se assim relevante, pelo menos num período transitório, explorar qual a melhor forma de retirar o máximo benefício do que a linguagem fornece de base, para a resolução dos problemas que um engenheiro de software enfrenta quando confrontado com a necessidade de desenvolver um sistema interactivo.

3. ETAPAS DO DESENVOLVIMENTO

O *Unified Process* (UP) não considera a interface como um sub-sistema com especificidades particulares, e genericamente incorpora-a nas fases de desenvolvimento. Deste modo, é criticado por não dar suficiente ênfase ao processo de descoberta dos requisitos de usabilidade e subsequente prototipagem e avaliação da interface.

Propomos, aqui, que a UML seja utilizada como mecanismo para a descrição da camada interactiva, e sugerimos alterações de processo ao delineado no UP, por forma a valorizarmos esta componente especial que merece um nível de detalhe razoavelmente mais complexo do que normalmente se lhe atribui.

3.1 Identificação das principais etapas

Por forma a utilizarmos a UML para a modelação da camada interactiva, e uma vez que não existe suporte nativo para este processo, é imperioso que se conjuguem os diagramas de forma a obter o efeito desejado. As alterações que se introduzem ao processo implicam que alguns diagramas são abordados mais cedo que o previsto no UP, visto que sendo assim auxiliam a descoberta de conhecimento sobre a camada interactiva. Apresentam-se agora as principais etapas e os diagramas utilizados em cada uma:

Análise de requisitos Embora no UP os requisitos derivados da camada interactiva estejam misturados com requisitos gerais de funcionamento, é possível registar todos os objectivos do utilizador que o sistema deverá suportar através de diagramas de Use Case. Torna-se necessário elencar, para cada um dos Use Cases, os cenários que o descrevem.

Análise de tarefas A descrição dos cenários, em que se conjugam requisitos e se estabelece uma ordem para a sua execução, pode ser utilizada como base para a modelação das tarefas. Alguns diagramas da UML, como os diagramas de actividade, são utilizados como ferramenta para a análise. Embora não exista um mapeamento linear entre linguagens de modelação de tarefas típicas, por exemplo a linguagem ConcurTaskTrees (CTT) [Patern? 99], e os diagramas de actividade da UML, é possível utilizar estes últimos para descrever o comportamento que suportará um determinado Use Case.

Desenho do diálogo A estrutura do diálogo que deverá suportar o comportamento descrito pelos diagramas de actividade anteriormente mencionados, pode ser representado em UML à custa de um diagrama de estados [Horrocks 99]. É nesta fase que se começa a definir a API que a camada da lógica de negócio deverá fornecer à camada de interface.

Desenho da arquitectura Através da descoberta, durante a fase de descrição do diálogo, do comportamento associado às transições, é possível descrever no diagrama de classes quais são as entidades da camada de negócio e qual o seu estado e operações. A partir desta camada o processo é muito semelhante à utilização da UML para a descrição de um qualquer sistema de software.

Modelação de comportamento Nesta fase descrevem-se, recorrendo aos diagramas de comportamento da UML, os diversos aspectos comportamentais do sistema; em particular o modo como os objectos cooperam para implementar as operações previamente identificadas.

Desenvolvimento e Implementação Onde se desenvolvem em código final as funcionalidade do sistema e se abordam os aspectos típicos de *deployment*.

Como tal, a linguagem de modelação não necessita de a suportar, no ponto de vista de quem modela, embora seja importante que a distância semântica entre estes dois níveis não seja muito grande.

4. UML NO PROCESSO DE DESENVOLVIMENTO

Como atrás referido, o nosso objectivo é a incorporação das ferramentas diagramáticas da UML no processo de modelação e desenvolvimento da interface. Como não existe suporte explícito para estas tarefas na linguagem, nem existe especial interesse no processo de desenvolvimento associado (o UP) pela componente correspondente à parte interactiva, o que propomos resulta numa metodologia de utilização da linguagem por forma a fornecer um suporte "não oficial" às tarefas identificadas.

Uma alternativa seria a construção de extensões à linguagem ou mesmo a construção de ambientes dedicados à modelação e especificação da interface. Contudo, apesar de esse ser um caminho mais directo, levaria a que o processo fosse apenas possível em ferramentas próprias, logo não estando de forma alguma incorporado no natural processo de modelação. Isso constituiria um risco na medida em que tal significa que efectiva e implicitamente estaríamos a reconhecer que de facto a camada interactiva e a camada de negócio constituem dois mundos separados.

Assim, as três fases da construção da camada interactiva que pretendemos incorporar são a *Análise de Requisitos*, a *Análise de Tarefas* e o *Desenho do Diálogo*. Para cada uma destas actividades, por esta ordem, é proposto um método de abordagem e utilização da UML, por forma a capturar a informação pretendida. Seguidamente descreve-se, de forma sucinta, o processo referido.

i) Análise de requisitos — utilizam-se os diagramas de Use Case para capturar os requisitos expressos pelos utilizadores do sistema. Este é o diagrama mais adequado para esta fase até pela definição de Use Case, que diz tratar-se de uma descrição informal dos requisitos funcionais, dos actores envolvidos e dos resultados produzidos.

Para melhorar a capacidade de expressividade da descrição associamos a cada um dos use case a descrição de um ou mais cenários, por forma a capturar a este nível toda a informação que se pode obter e que é importante para a camada interactiva. Requisitos de usabilidade, tal como outros requisitos não funcionais, podem ser incluídos, nesta fase, como notas ou restrições ao modelo.

ii) Análise de tarefas — utiliza-se a informação obtida anteriormente para extrair a informação necessária à especificação das tarefas.

Abordagens como as CTT capturam formalmente esta descrição, mas em UML podemos recorrer aos Diagramas de Actividade por forma a garantir um nível de expressividade semelhante.

Recentemente foi acrescentada aos diagramas a capacidade de delinear regiões interrompíveis (no que concerne ao controlo de fluxo), o que permite a especificação de operações de cancelamento da tarefa. Falta ainda, em

relação à expressividade oferecida pelo CTT, a capacidade de certos diálogos poderem ser temporariamente suspensos e posteriormente retomados (operador |>). Não é ainda linear a capacidade de efectuar estruturação e encapsulamento explícito de diagramas de actividade, pelo que a descrição da análise de tarefas obtida necessita por vezes de documentação de apoio.

iii) Desenho do diálogo — nesta fase, através da informação existente nos Diagramas de Use Case e de Actividade, identificam-se os ecrãs necessários à aplicação e, em função das descrições existentes nos Diagramas de Actividade desenvolvem-se Diagramas de Estados. Os diagramas de estados assim obtidos, não apresentam, a nível de fluxo de informação, alterações significativas em relação ao correspondente diagrama de actividades. No entanto, o foco de atenção passa das actividades realizadas por cada actor, para o impacto que essas actividades têm na interface.

Ao descrever as transições torna-se também possível decorá-las com os métodos que a camada de negócio deverá disponibilizar para suportar o comportamento descrito. Este passo metodológico permite que se já refinando o modelo e que sejam levantados os métodos da camada de negócio que implementarão as acções associadas à camada de apresentação.

Sendo este processo nitidamente iterativo, através da descrição dos diagramas de transição de estado vai-se completando a informação sobre o sistema e, além de se especificar a componente relativa ao controlo de diálogo, também se vai adquirindo informação sobre a arquitectura da camada de apresentação.

Descritas as fases da modelação da interface e sua correspondente incorporação na linguagem UML, importa referir algumas pré-condições que regem o método proposto: a) existe um diagrama de actividade para cada use case descoberto na análise de requisitos (cada diagrama de actividades deve conter toda a informação dos possíveis cenários que se podem obter a partir de um use case); b) deve existir um caminho no diagrama de estado para cada linha (fluxo) que se possa traçar no diagrama de actividades; c) o conjunto dos métodos obtidos na decoração dos diagramas de transição de estados para uma determinada entidade representa o comportamento exibido por essa entidade (logo, não existe comportamento numa entidade que não seja obtido a partir do processo de construção do diagrama de transição de estado).

A derivação da lógica de negócio é obtida através da completude do processo de construção dos diagramas de transição de estado. Apesar de este processo ser nitidamente iterativo e ser difícil determinar quando se chega a uma situação estável, o processo defendido neste artigo produz uma substancial parte do comportamento que constitui a lógica de negócio. Parece ser também a forma mais natural de determinar a lógica de negócio, uma vez que os métodos obtidos por este processo derivam totalmente da análise das tarefas que foram identificadas. Sendo assim, o comportamento obtido corresponde à totalidade (ou quase)

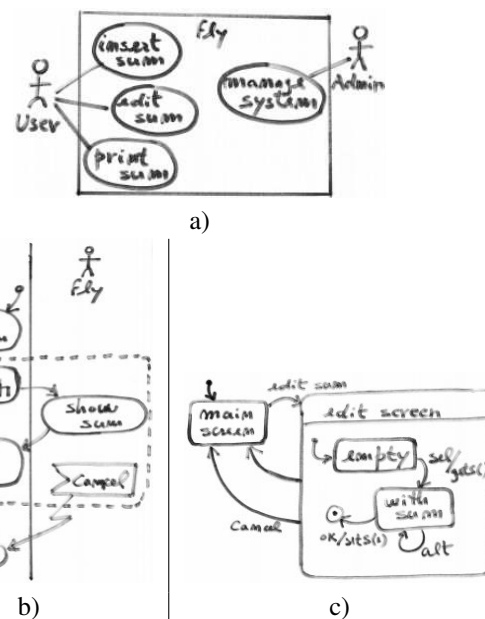


Figura 1. Diagramas UML

do que o sistema deve exibir. Existirão sempre métodos utilitários, ou resultantes de algum processo de *refactoring* ou re-ordenação do código, mas a componente principal que corresponde aos métodos da interface deverá ser encontrada através do processo defendido neste artigo.

5. UM EXEMPLO

Para cada fase mencionada na secção anterior, iremos agora apresentar (através de extractos de um exemplo — o sistema de gestão de sumários *Fly*) de que forma os diagramas UML podem ser utilizados numa perspectiva de desenvolvimento centrada no utilizador.

5.1 Análise de requisitos

Tal como referido, recorrendo a diagramas de Use Case, identificam-se os utilizadores do sistema, os seus objectivos na utilização do mesmo e os objectivos de usabilidade. A figura 1.a) apresenta um extracto do modelo. Nela podem ser identificadas duas classes de utilizadores (actores): Users e Admins. A cada actor pode estar associada informação relativa a características consideradas relevantes, como nível de perícia ou outras. É importante realçar que, através do mecanismo de especialização/generalização, é possível realizar *user profiling*.

5.2 Análise de tarefas

Uma vez identificados os objectivos dos utilizadores, torna-se agora necessário descrever qual a interacção necessária para atingir cada um deles. É importante desenvolver um modelo de tarefas que permita, não só raciocinar sobre o sistema, como guiar a sua implementação. Na figura 2 apresenta-se o que poderia ser uma primeira abordagem à modelação da tarefa relativa ao objectivo Editar Sumário. O que o modelo descreve é um conjunto de comportamentos válidos.

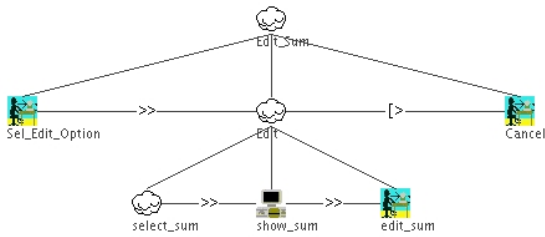


Figura 2. Modelo de tarefas em CTT

Na figura 1.b) apresenta-se o mesmo modelo de tarefas, desta feita descrito utilizando Diagramas de Actividade. Note-se a utilização de regiões interrompíveis para modelar o comportamento de cancelamento do diálogo e a utilização de pistas para associar as acções (actividades) aos diferentes actores. Neste aspecto, os Diagramas de Actividade apresentam alguma vantagem em relação às CTT, pois permite ilustrar a interacção do sistema com vários utilizadores em simultâneo de forma simples e directa. No entanto, a estruturação pretendida em sub-objectivos perde-se em grande medida. A única forma de a manter seria utilizar sub-actividades (ver actividade **Select Sum**) com o que isso implica de utilização de sub-diagramas, causando uma maior dificuldade na obtenção de uma visão global da tarefa.

5.3 Desenho do diálogo

Uma vez estabelecido um modelo de tarefas, torna-se necessário desenvolver uma interface que o suporte. Centrarmos-nos na modelação do diálogo, uma vez que, cada vez mais, os IDEs fornecem suporte de qualidade para a modelação da componente gráfica. Assumindo uma interface típica organizada em écrans, o objectivo passa por identificar todos os écrans necessários a uma dada tarefa e qual o controlo de diálogo necessário para a suportar.

Na figura 1.c) apresenta-se o modelo para a tarefa anterior. O modelo identifica, não só os diferentes écrans da aplicação, como situações relevantes desses écrans. O controlo de diálogo é modelado através das transições entre estados. Estas transições permitem ainda definir quais os métodos da camada de negócio necessários à implementação da interface. O conjunto de todos os métodos derivados define a API que a lógica de negócio deverá implementar. Deste modo, esta fase constitui-se como a ponte para o desenvolvimento dessa camada aplicacional.

6. DISCUSSÃO E CONCLUSÕES

Apresentou-se neste artigo uma primeira abordagem à utilização da UML no desenvolvimento de um sistema interactivo. É proposta a utilização de diferentes diagramas da linguagem UML por forma a fornecer suporte a algumas das preocupações de um desenvolvimento centrado no utilizador e ilustra-se a abordagem com extractos de um exercício de modelação e desenvolvimento. A abordagem é complementar aos esforços de introdução de melhoramentos na linguagem, na medida em que procura explorar que o já existe por forma a avaliar as potencialidade e falhas da norma em vigor.

A abordagem apresentada é ainda preliminar, mas as novas características da UML 2.0 apresentam um potencial interessante. Até à data a validação da abordagem tem sido efectuada através da sua aplicação à modelação/desenvolvimento de um número limitado de aplicações de pequeno/médio porte. Pretende-se continuar a análise das suas vantagens/desvantagens através da sua aplicação a sistemas de maior envergadura. Pretende-se ainda avaliar a possibilidade de realizar raciocínio sobre os modelos por forma a validar a correcção dos diferentes modelos que vão sendo desenvolvidos.

7. AGRADECIMENTOS

Este trabalho foi suportado pela FCT (Portugal) e FEDER (União Europeia) através do contrato POSC/EIA/56646/2004.

Referências

- [Group 05] Object Management Group. Unified modeling language: Superstructure, v. 2.0. OMG specification formal/05-07-04, August 2005.
- [Horrocks 99] I. Horrocks. *Constructing the User Interface with Statecharts*. Addison-Wesley Professional, 1999.
- [Jacobson 99] I. Jacobson, G. Booch, e J. Rumbaugh. *The Unified Software Development Process*. Addison-Wesley, 1999.
- [John 03] B. E. John, L. Bass, e R. J. Adams. Communication across the HCI/SE divide: ISO 13407 and the Rational Unified Process. Em *Proceedings of HCI International 2003*, June 2003.
- [Krasner 88] G. E. Krasner e S. T. Pope. A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49, August/September 1988.
- [Nunes 01] N. J. Nunes e J. Falcão e Cunha. Wisdom — A UML based architecture for interactive systems. *Lecture Notes in Computer Science*, vol. 1946, 2001.
- [Patern? 99] F. Patern?. *Model Based Design and Evaluation of Interactive Applications*. Applied Computing. Springer Verlag, 1999.
- [Patern? 00] F. Patern?. ConcurTaskTrees and UML: how to marry them? Position paper at TUPIS'00 – a UML 2000 Workshop. York, UK, October 2000.
- [Silva 02] P. P. Silva. *Object Modelling of Interactive Systems: The UMLi approach*. Tese de Doutoramento, Department of Computer Science - University of Manchester, 2002.

Formalizing Markup Languages for User Interface

Luis G. M. Ferreira
Grupo Sistemas de Tecnologia e Informação, IPCA-EST
Barcelos
lufer@ipca.pt

Resumo

This work focus on the formalization process of user interfaces specification using eXtended Markup Language (XML) description. As an instance of this process, we explore a VDM-SL specification of UIML - User Interface Markup Language. The main results are tested in a particular graphical tabular OLAP features case study, towards a definition of a Visual Component Library, with UI components composition and reuse.

Palavras-Chave

User interfaces, XML, VDM-SL, Formal methods, OLAP.

1. Introduction

For the companies of today it is crucial the full integration and consistency of all the information which flows or is stored in its databases. The number of specific and different applications, manipulating this information, is necessarily large. Maintenance and support of these applications requires expensive team work.

The (conventional) *Relational Database Management Systems (RDBMS)* no longer can guarantee timely efficiency in the answers to complex queries. The treatment of information in bidimensional format (tables and spreadsheets, for example), the inability of the analysis, transformation and consolidation of such information, restrict the overall application of these systems.

The advantages of formal specifications of complex systems using formal methods are well known. It is known and accepted by a considerable group of persons as a way to avoid ambiguities and clearly guarantee correctness and consistency of the developed programs. In this context, it is useful to experiment these guaranties on user interface development, exploring some kind of merging processes of new specification methods with existing graphical objects specifications.

As a consequence of this large set of graphical elements, many of them only found in pure commercial application, its formal specification process, towards a *Visual Component Library - VCL*, could become complex. This complexity can even increase during the tentative abstraction of real problems.

This paper is organized as follows. Section 2 presents the motivations and main technical guidelines followed on this work. This is followed, in Section 3, by the brief presentation of - at time of writing - technologies on modelling,

specifying, designing and programming user interfaces, as well as OLAP concepts and Markup Languages applied to interfaces description. Section 4 discusses the archived UIML VDM-SL specification, followed by resultant table model and its specification on VDM-SL on Section 5. Section 6 describes the developed prototype and main supporting tools. Finally, Section 7 concludes the paper and foretell future developments.

2 Problem Statement

This work stands from the thoughts of Brad Myers [Brad 98], where the area of the UI is an example of the strong influence of academic research in the industrial way of doing things. In its essence, this research intends to apply formal semantics techniques to the user interface development, in order to impose scientific rigor on the whole process, from interface specification and validation to its transformation ("transcoding"). However, the application of formal methods should never surpass the limits of versatility, trying whenever possible to build the specification through the composition of components previously specified.

By applying formal methods, UI development becomes a rigorous discipline with focus on higher abstraction levels (relative to implementation), using visual components such as *ListBox*, *Button*, *TextBox*, *Menu*, etc.) and a semantic set of rules which describes the process.

The *Sets Theory* [Oliveira 03] semantic specification of each component will resort to semantic models, describing their distinct internal states, associated to execution of each operator.

The process usually starts from a data model, properly (or not) normalized, to describe the intended information system. Using *VDM-SL*, developers will try to specify its data

model, by identifying the interface components, necessary for its information visualization. To guarantee coherence in the final placement of the components in the interface layout, it should be interesting the use invariants which, in the later phase of layout placement, should prove its relative position.

3. State-of-the-Art

This work, being specially focused with data Visualization in application interfaces, gave priority to the analysis of the existing technologies in the area. In [Phanouriou 00], *Constantinos* summarizes the main steps that can be identified in this process and, as the *Seeheim model* [Pfaff 85] advocates, the importance on maintaining a real separation between the presentation layer and the remaining application layers.

Amongst the known user interface Models, from the "ancient" to the recent ones (*Arch Model*, *MVC Model*, *PAC Model*, *XForms*, *N-Tier Model*, *MIM Model*[Phanouriou 00]), perhaps the first and the most significant UI model was *Seeheim Model*, presented at the Seeheim Workshop (Berlin, 1985). As this model describes, as well as on recent architecture models (such as *N-Tier*), there are evident concerns in separating responsibilities between interfaces and the rest of the application.

The *MIM -Meta-Interface Model*, which extends the level of abstraction of the *Slinky Model*, was created with abstractions mechanisms to describe interfaces that can map into multiple and distinct types of devices. It divides the interface into three major components: *presentation*, *logic* and *interface*.

The *UIML*, markup specification language for generic interfaces description explored on this work, is based upon *MIM* model.

3.1 Multidimensional Analysis

Once we intend to explore OLAP features applied to tabular object as a case study, the Multidimensional Analysis technique over multidimensional information was carefully explored.

Multidimensional data models categorize data as being either *facts*, which means data values and eventual attributes, or as being *dimensions*, that categorize the facts and are mostly textual. The model of a Multidimensional Database is an array of n-dimensions (often called *Hypercube* or collection of related cubes). Although the term "cube" implies three dimensions, a cube can have any number of dimensions.

Each dimension is associated to a hierarchy of data levels consolidation. For example, a dimension "time" can have a hierarchy with levels *day*, *month* or *year*. A dimension works as an index of identification of values in the array. Each array position, corresponding to the intersection of all dimensions, is a *cell*.

Operations like *Rotation*, *Ranging*, *Rolling-Up*, *Drill-Down*, *Hierarchies*, etc. should be available, supporting OLAP technologies.

3.2 Markup Languages for User Interface description

Over the past few years, there have been a number of industrial and academic initiatives to standardize many data types towards application interoperability. However, it seems that the same did not happen in the interface design area.

It is relieving to know that the *W3C - World Wide Web Consortium* and *OASIS - Organization for the Advancement of Structured Information Systems* have worked towards the great dissemination of XML and of some standards for application interoperability (like SOAP¹) and several XML applications for multiple distinct areas. Focusing on our main goal, there are already several XML applications aiming user interface specifications².

Noting the scope, requirements and structure of *XIML - eXtensible Interface Markup Language*, *XUL - XML-based User-interface Language* (by Mozilla project), *AUIML - Abstract User Interface Markup Language* and *UIML - User Interface Markup Language*, we focus our attention on the last one, mainly because its extensibility and facility.

We can also confirm that *UIML* does not walk alone on this path, ignoring all other initiatives. For instance, in the current *UIML* version (3.0) it is even better to use *XForms* with *UIML* than to work directly with *HTML*.

4 UIML Formal Specification

The *UIML* should not be considered a pure VCL because it "works" away from any concrete representation. However it can be understood as so in the sense that it describes visual components that can be represented in a concrete graphical environment.

All *UIML* elements were formally specified. `<uiml>`, `<head>`, `<structure>`, `<peer>`, etc., using VDM-SL notation as following code excerpts try to represent.

The `<uiml>` specification was:

```
Uiml :: head: [Head]
        members:Member*
inv uiml = uniqueIds(uiml) and
validProperties(uiml);
```

where *inv* represent the invariant which must be respected.

The `<interface>` element specification was,

```
Interface ::   intele: InterfaceElements*
               id: [ID]
               source: String
               how: [SourceModes]
               export: [ExportOptions]
inv i = uniqueIDs(i);
```

The `<structure>` element specification was,

¹SOAP - Simple Object Access Protocol

²<http://xml.coverpages.org/userInterfaceXML.html>

```

Structure :: parts : Part*
           id: [ID]
           source: String
           how: [SourceModes]
           export: [ExportOptions]
inv s = uniqueIDs(s);
    
```

In order to certify semantic rules on operators utilization, several VDM-SL *invariants* were created, as described in the code excerpts. The code below represents an excerpts of one of such definitions.

```

UniqueIDs: UIMLElements -> Bool
UniqueIDs(t) =
  case t:
  mk-UIML(-,-) ->
    rng allIDs(t)={1},
  mk-Peers(-,-,-,-) ->
    rang peersIDs([t],{->})={},
  ....
    
```

All the VDM-SL specifications respect the actual UIML XML Schema[Harmonia 02]. Due to the extension of UIML language, some particularities were not considered on this work, mainly rules associated to *template* and *property* elements.

The resultant VDM-SL specification, will be tested using a table object.

5 Table VDM-SL specification

To better understand OLAP methods behavior (functions, operators or transformers), they must be considered as **attributes** if they support column and row characteristics (ex. *column name*, *column width*, etc.); **operators** if they work with table structure (*addRow*, *delRow*, etc.); **functions** if they support data calculus (*sum*, *avg*, *sort*, etc.) or even **transformers** if they manipulate the original table structure (*hideCol*, *drill-down*, *rotation*, etc.).

Following excerpts of VDM-SL code specify some implemented table operators. For instance, the operator responsible from table creation - *mkTable* - was defined as:

```

mkTable : Style x Rows -> T
mkTable (s,r) = mk-(s,r);
    
```

For example, the *dT* default table creation, with no rows and no graphical information (background color, cell spacing, etc.), could be obtained by the following expression:

```
mkTable("dT", {}, {| ->});
```

Another table *t* could start its construction using:

```
mkTable(, "r1" | -> mk-(, "Mark" | -> mk-(, "Ford")));
```

The *hideRow* operator, which hides a particular row:

```

hideRow : T x Rid -> T
hideRow (t,rid) =
  mk-(t.#1,markRow (t.#2,rid,"h"))
pre rid 2 dom (t.#2);
    
```

The *delRow* operator, which delete a particular table row,

```

delRow : T x Rid -> T
delRow (t,rid) = mk-(t.#1,{rid}<-t.#2);
    
```

In our example, the expression

```
delRow(t, "r1")
```

will remove all *Black Ford* information concerning.

A similar process was done for all others implemented operators.

6 Prototype and Supporting Tools

In order to experiment the animation of VDM methods, mainly *OLAP* functions, we have decided to create an HTML prototype whereby we can visualize all table transformations. This application uses our VDM specifications, *UIMLSpec* and *UIMLSpecTab*. From CGI HTML forms behavior, the VDM-SL methods are called, then UIML is generated and rendered again to HTML. Figure 1 depicts the prototype architecture.

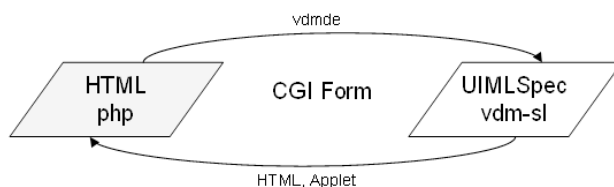


Figure 1. Prototype architecture

As supporting tools used on the four phases of the process, on Phase 1 - *Transcoding UIML to VDM-SL*, we generated a XML StyleSheet (*uiml2vdm.xml*) to transform UIML into VDM-SL. Next we created directly in VDM-SL, a Pretty-Print tool - *vdm2uiml.vdm* - which works as a script and executes a parsing process over VDM-SL specifications. It generates a correspondent UIML syntax for each specified element.

In Phase 2 - *Verifier*, was created a verifier to support the *validation* phase of our process. This validation allows to test the consistency of the VDM-SL generated code along the transcoding process, verifying the "similarity" between both UIML codes.

The Phase 3 - *Abstraction*, has not been completely developed. All reasoning implicit in it is sketched in Future Work. The process starts from VDM-SL table abstraction towards the creation of a visual components library.

This work should show that it is possible to perform transformations by calculation. Then it is necessary to find different candidate implementation objects. This is a natural property of adaptable interfaces.

In our case study, the idea stays as "it is possible to get a new UIML interface description" which represents the same.

In Phase 4 - *Rendering UIML*, we used the actual UIML rendering engines, from Harmonia³, which supports

³<http://www.harmonia.com>

several platforms, with several programming languages (JAVA, HTML, WML, VoiceXML, etc.), rendering widgets and events specified in *UIML* <presentation> elements. The goal was to generate UI from an *UIML* specification.

For instance, in order to convert our table specification (*UIMLSpecTab*) to our base specification (*UIMLSpec*), we created the function *TabUIML2UIML*. Considering this, the expression

UIMLSpecTab' *TabUIML2UIML*(*UIMLSpecTab*' *t*)

will result in an *UIML* element corresponding to our table *t*.

Once in the *UIML* code, the following shell commands will respond for the rest of the process:

1. *u2h table.uiml table.html*
Generates *HTML* code for browser platforms. The render *u2h* works over *HTML* vocabulary.
2. *u2ji table.uiml table.java*
Generates *Java* code associated to *UIML* descriptions, ready to be used on Java platforms. The *u2ji* render works over *Java* vocabulary.
3. *u2w table.uiml table.wml*
Generates *WML* code for *WAP* devices. The render *u2w* works over *WML* vocabulary.

7 Conclusions and Future Work

The main focus of this work is the user interfaces graphical design which support current software applications. For the sake of rigor, it addresses the application of formal methods to recent UI markup language support technologies. The application of formal semantics techniques to user interface development, imposes scientific rigor on the whole development process: specification, validation and transformation.

Despite the existence of several research projects focused on this, there are still questions which remain unanswered. One of these questions is as basic as follows: are markup languages, even developed with important objectives, the best idea to support this process? Further than a portable and platform independent description technique, XML subscribes to an enormous vocabulary set, which demands parser and validation mechanisms.

Our *UIML VDM-SL* specification leads us to believe that there is room for some conceptual simplification. This conclusion is sustained by observing the achieved abstract specification, where several replicated elements and repeated patterns can be identified.

Considering this, an interesting path for future work is that of *UIML*'s refactoring process towards a simpler specification.

Another interesting focus topic for future developments is that of creating a formal *Visual Component Library - VCL*. Our table specification is but the beginning of one such repository of useful graphical interaction objects (IO). The aim should be the construction of new interfaces by reusing existent components. Once made available, a VCL would contribute to answering another quite common question: "What is the best IO to represent this particular piece of data?", Ideally, a "matching" process should become available for searching the library and retrieving all adequate components.

As a matter of fact, our research has included some work in these two vectors - *UIML* abstract syntax refactoring and visualization. However, the outcome cannot be regarded as finished work. Despite their incompleteness, perhaps our results can still be useful to anyone wishing to pursue them.

8. ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor José Nuno Oliveira, member of DI (Department of Informatics, Minho University⁴), who encouraged all formal methods research and initiatives at the University, for his useful support and advice during this work. I am also grateful to Marc Abrams and James Helms *Harmonia* members, for their enthusiastic support along this research as well for the availability of their *UIML* supporting tools.

References

- [Brad 98] Myers Brad. A brief history of Human Computer Interaction Technology, 1998. <http://citeseer.nj.nec.com/myers98brief.html>.
- [Harmonia 02] Inc. Harmonia. User Interface Markup Language UIML Specification. Technical report, Harmonia Inc., 2002.
- [Oliveira 03] J. Nuno Oliveira. An introduction to Data Refinement. Formal Methods II, 2003.
- [Pfaff 85] G.E. Pfaff. User Interface Management Systems: Proceedings of the Seeheim Workshop, 1985.
- [Phanouriou 00] Constantinos Phanouriou. UIML: A Device-Independent User Interface Markup Language, 2000.

⁴<http://www.di.uminho.pt>

On the road to Rich Digital Books

Rui Lopes Carlos Duarte Hugo Simões Luís Carriço
LaSIGE, Faculty of Sciences, University of Lisbon
Edifício C6, Campo Grande, 1749-016 Lisboa, Portugal
{rlopes, cad, hsimoes, lmc}@di.fc.ul.pt

Abstract

This article presents an architecture for producing rich digital books. These are books enriched with supporting media, like images and sound, to reach broader audiences and enlarge their usage possibilities. The architecture is based on automating the books' production process, increasing their availability. It also enjoys a high flexibility degree, allowing the production of books that can be read either with off-the-shelf software or specialized digital book players. One such player, capable of meeting the demands of active reading is also presented in this article.

Keywords

Rich Digital Books, Automated Book Production, Behavioral Dimensions, Adaptive Multimodal User Interfaces

1. INTRODUCTION

Before digital libraries, several barriers were in the way of everyone's right to information. Today, users can read books in the comfort of their homes, thus another level of issues becomes relevant. The blind, children, students, and the average user should have access to information. However, current technologies are unable (or become too expensive) to cope with this. Books should be enriched with new information, new media, or even rewritten. As it is impossible to have an *one size fits all* solution, different interfaces must be provided by reproduction platforms.

With this range of possibilities, manual production of rich digital books will be time consuming and error prone. Therefore, production should be as automated as possible, driving the focus of manual tasks to specialized activities (e.g. describing content's metadata). This way, book collections can be delivered to a wider audience, through diverse usage scenarios. General-purpose time based hypermedia frameworks [vO01] may produce similar results, but lack critical text-based formatting capabilities.

This article presents an architecture for rich digital books. Different reproduction platforms can be targeted by production processes, e.g. limited players or powerful players with adaptive capabilities. Such players are presented in the article, accomplishing the same goals that a set of dedicated digital reading platforms have achieved [Schilit 99], as current playback devices [DC] suffer from severe usability and accessibility flaws [Duarte 05].

2. REQUIREMENTS

Reading activities are greatly influenced by readers' goals, resulting on different commitment and attention levels. To portray the diverse kinds of reading, a categorization of

reading situations was proposed in [Schilit 99]. Two dimensions – nature of engagement (passive or active reading) and breadth of the activity (single or multiple texts) – are used to categorize four conventional reading situations [Schilit 99]. While understanding the text is the goal of all situations, each one has different problems. Multiple texts entail the need to manage multiple documents and the difficulty on finding the needed information. Active reading [Adler 72] involves underlining, highlighting and annotating, either on the text or in a separate notebook, demanding an annotations management mechanism.

Digital books and digital libraries may contribute to mitigate some of these problems. Digital books offer the possibility to record, organize and search annotations entered by the reader [Schilit 98]. Digital libraries make it possible to manage book collections, and to create and explore relations between books. The possibility of sharing personal annotations within the community of members of a digital library conveys a social sense to the books [Kaplan 05].

Besides these affordances, the book's digital support opens allows for enriching its content with supporting media [Carriço 03]. This way, the book can be enriched with new multimedia content. In such scenario, the book content can also be narrated in addition or in alternative to the visual presentation, similar to Digital Talking Books [AN02].

The rich digital book in a digital library context must cope with a heterogeneous audience and a variety of situations: Researchers, students and other professionals who can benefit from the advanced navigation features, annotation support, and the integration with a large collection of related material; Pleasure readers in constrained environments can benefit from the alternative modes of presentation. An audio-centered book can be used in situations

where visual focusing is more difficult or even impossible; Both children and adults can enjoy visual and audio enhancements that enrich and complement the book's presentation; Children and other reading/writing disabled persons benefit from simultaneous narration and visual presentation when learning how to read/write; The blind can benefit from audio narration, coupled with navigation mechanisms (e.g. table of contents), and speech-based annotations; Partially sighted users will benefit from the enhancement of visual components. Therefore, books' contents (and its navigation, presentation and interaction mechanisms) should be tailored according to each situation.

These issues have an impact not only on the reproduction platform, but also directly on the book's characteristics. To be able to meet all the proposed goals, the book's format must be presentable in some reproduction platform, and enriching materials must be selected according to the book's content, allowing book authors and publishers to easily create different editions. Also, through production automation based on a normalized digital library, the capabilities offered by reproduction platforms will be met.

To fit all these aspects in a rich digital book creation process, production architectures must be modular, easing both book creation and maintenance tasks, as follows: provide modular and composable content processing configuration; define a rich content format supporting multimedia composition; allow the addition of content to a central repository; support addition of new material to the content; provide a clear separation between the book's content and its UI; and define a reusable specification of UIs, enforcing coherence amongst different usage scenarios.

Having flexibility in the book production process raises issues regarding production time users. As such, three user profiles must be supported: top level user, power user, and developers. The first is typically a user with little technical expertise (e.g. digital library maintenance staff), whose tasks relate to specify and annotate content, and control profile-based batch production. The second profile relates to those who have high knowledge over digital publishing and, therefore, should have full control of book production processing steps, to create production profiles (geared towards publishers' specificities). Lastly, developers are specialized in creating digital publishing components.

In order to support the different rich digital books scenarios, a multimedia repository must be available both at production and reproduction times supported by the following: continuous identification, classification, and organization of multimedia items; converting, and structuring data into a normalized format; establish relationships between media items, based on different criteria (e.g. semantic or composition based); lastly, support online query and retrieval of multimedia items, to be integrated into manual and semi-automatic content production tools.

3. RICH DIGITAL BOOKS PRODUCTION

A flexible architecture for automated production of rich digital books was defined, based on the requirements pre-

viously gathered (see Figure 1). An input is fed to production (by top level users), where it is transformed, augmented, and/or simplified according to a production profile (e.g. based on publisher's requests). At the end, a tailored rich digital book is delivered. As documents are specified in (or transformed into) XML languages in both ends of the architecture, the document flow was implemented with XML technologies (created by domain expert developers).

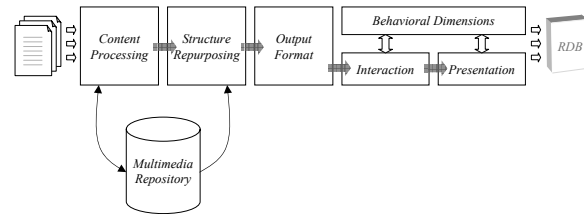


Figure 1. The production architecture.

3.1. Content Processing

The increase on the production and use of rich contents requires an effective, efficient, and reliable multimedia content management. To handle these issues, the processing architecture's first concern deals with different tasks centered on book content processing. As several input formats may be fed, a normalization task is performed initially.

Afterwards, content reasoning tasks should be performed. These can be classified as manual, semi-automatic, or fully automated, depending on the content's richness and complexity (semantic analysis vs. automated syntactic analysis). These tasks are supported by a multimedia repository through its rich modeling capabilities and content based multimedia storage, indexing and classification, to facilitate on retrieval and integration facilities (see Figure 2).

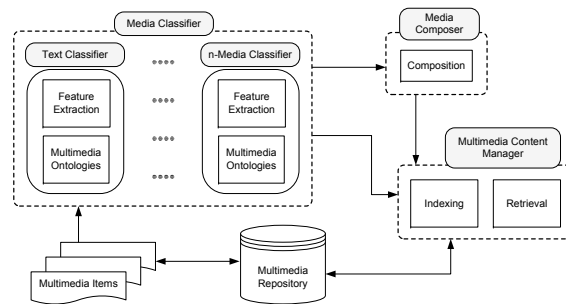


Figure 2. Multimedia items classification.

3.2. Structure Repurposing

The second concern relates to content structure repurposing. Initially, different tasks provide powerful content reasoning features, enforcing content reuse mechanisms and multimedia repository enrichment with the current state of the rich digital book's content. Afterwards, structure extraction tasks can be applied to the normalized content. A simple example is the extraction of a table of contents as independent structured content modules. Lastly, some

control over content structures can be performed, regarding their complexity (e.g. if targeted to low-end devices).

3.3. Output Format

The third concern in the production architecture relates to output format conversion. This concern defines tasks for the conversion of normalized content structures. As different scenarios must be taken into account, different formats are supported (e.g. HTML+TIME). Richer formats allow the integration of interaction and presentation capabilities, whereas more limited platforms require simple content formatting. Afterwards, different tasks can be applied to the current processed book content state, integrating user custom constructs (such as skeleton structures for bookmarking and annotating), if applicable within the chosen format.

This concern provides the minimum output to be played on a rich digital book reproduction platform, as some platforms are rich enough to provide flexible interaction and presentation capabilities. Consequently, the specificities for these two concerns are optionally applied later on.

3.4. Interaction and Presentation

After the playback platform is chosen, and the book's content is transformed into a specific output format, behavioral dimensions are introduced in the production architecture. These dimensions define how a reproduction platform should handle interaction and presentation issues over a book's content. Corresponding concerns are introduced in production time to handle playback platforms' limitations.

The first concern relates to interaction. If the output format allows its configuration, tasks implement interaction mechanisms for specific inputs (e.g., mouse, speech). Interaction limitations are also handled (e.g. limiting speech recognition vocabularies in crowded environments).

The other concern defines how a book is presented. To ease configurability and keep UI coherence amongst different output formats, the architecture defines presentation profiles (sets of presentation rules), applied to the book's content. By combining different rules, profiles share presentation features (enforcing UI coherence). Also, by selecting richer presentation profiles, stricter behavioral dimensions are fed to the reproduction platform.

4. REPRODUCING RICH DIGITAL BOOKS

The production process flexibility outputs digital books tailored to a reader's desired format. This means that books can be available for immediate presentation on several devices or platforms, or that the production process output can be fed into a dedicated book reproduction artifact that will deal with the interaction and presentation aspects.

In the first case the production outputs ready-to-read books on commonly available presentation platforms. The output formats currently available in our production platform include print formats and other ones combining text and audio, like SMIL (which can be viewed with RealPlayer), or HTML+TIME (using Internet Explorer). Other formats, targeted at more specific user populations, may also be generated (e.g. a Braille version for the blind).

However, no book player meets all the requirements elicited. These requirements, namely the possibility to present the same content with different media and be operated by visually impaired users, demand for multimodal interaction handling. The several personalization capabilities that must be available, and the need to manage possibly simultaneous media streams, also suggest the introduction of adaptive capabilities. To meet these requirements, a player with adaptive multimodal capabilities was developed, based on the FAME framework [Duarte 06].

For rich digital books reproduction four components were identified: main book content, table of contents, annotations and supporting media. A set of dimensions is identified in each component, describing how they relate to each others during the book's reproduction, and also determine the way each component is presented and how the user can interact with it. For the mentioned components, the identified dimensions allow the adaptation of the following aspects of presentation and interaction: modalities used to present content and receive instructions; synchronization granularity unit between audio and visual presentation; audio narration speed; text that is presented marked based on attached annotations or media items; reading path followed based on reader-made annotations; component visibility; intrusion and awareness raising features of the player whenever there are supporting contents or annotations to be presented; narration behavior whenever supporting contents or annotations are presented; and Table of Contents and annotations window presentation.

The developed player (Figure 3) improves the reader's situational awareness: a marker highlights the excerpt being narrated; the current section or chapter number is highlighted in the Table of Contents; and, the current reading position is afforded by the main content scrollbar. The current section or chapter number can be narrated at anytime, providing the information for the blind. To perceive the book's structure, the user can scan the Table of Contents at anytime without dismissing the narration.

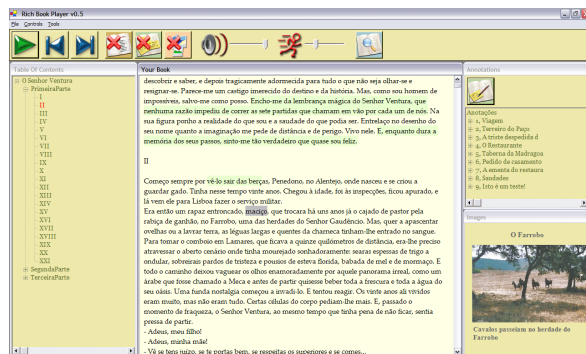


Figure 3. Rich digital book player.

Advanced annotation support offers the reader the possibility to create and manage bookmarks and notes. Raising the reader awareness to annotations during the book's presentation is achieved in two distinct ways: the show/hide annotations button flashes every time the narration reaches

a point in the text that has been annotated and the annotations window is not visible. If the annotations window is displayed, the text that has been annotated is highlighted in the main content window. The awareness to supporting media is raised similarly. Audio alerts are also provided to signal the presence of an annotation or an image.

The player supports active reading of books, by allowing the reader to annotate the text and categorize the annotations. From these categories, several behaviors can be devised for further readings of the same book: reading of the annotated material; reading material of specific categories; and associate different reading speeds to different categories. This allows user creation and reading of trails that may constitute sub-stories, argumentation paths, etc.

In order to minimize user interventions, the player can adapt its behavior regarding presentation of annotations and other supporting material. If alerting the user to the presence of images or annotations is usually met with positive feedback, and the user decides to view the image or annotation, then the player will update its behavior to showing the image or annotation, instead of alerting to its presence. But, if the user chooses to ignore the alerts, the player will then update its behavior to stop alerting to the presence of images and annotations.

5. CONCLUSIONS

To increase the availability of digital books in general and DTBs in particular, it is essential that the production process moves from the manually assisted process that it is today, to a mostly automatic one. This article presented a rich digital book production architecture to move us closer to such a vision. With the automation of the production process, the architecture enables goals such as providing the same *brand* for a whole collection offered by a digital library, or preparing special editions of books targeted for specific audiences. To allow for the production of such books, the architecture is defined by mechanisms to normalize content, perform structure repurposing and define output formats. The preparation of special editions is made possible by the close integration with a multimedia repository, which provides contents to enrich and complement the book. During the production process, contents are added to the repository for future uses.

The described architecture proved its flexibility by producing books that can be read in readily available tools, or targeted at specialized digital book players. In the first case, output formats encompass print books, presentations combining text with narrations, and even Braille. Specialized players can offer features like advanced navigation, or annotation management and sharing that allow active reading practices, and increase the possible spectrum of readers and reading situations. One such player, developed to meet these goals, was also presented in this article.

6. ACKNOWLEDGEMENTS

This work is funded by Fundação para a Ciência e Tecnologia, through grant POSI/EIA/61042/2004, within the context of project RiCoBA (Rich Content Books for All).

References

- [Adler 72] Mortimer J. Adler and Charles Van Doren. *How to Read a Book*. Simon and Schuster, New York, 1972.
- [AN02] ANSI/NISO Specifications for the Digital Talking Book, 2002. Available at <http://www.niso.org/standards/resources/Z39-86-2002.html>.
- [Carriço 03] Luís Carriço, Nuno Guimarães, Carlos Duarte, Teresa Chambel, and Hugo Simões. Spoken books: Multimodal interaction and information repurposing. In *Procs. of HCII'2003, International Conf. on Human-Computer Interaction*, pages 680–684, Crete, Greece, 2003.
- [DC] Daisy consortium playback tools. Retrieved January 18, 2006 from <http://www.daisy.org/tools/playback.asp>.
- [Duarte 05] Carlos Duarte and Luís Carriço. Users and usage driven adaptation of digital talking books. In *HCII '05: Procs. of the 11th International Conf. on Human-Computer Interaction*, Las Vegas, Nevada, USA, 2005.
- [Duarte 06] Carlos Duarte and Luís Carriço. A conceptual framework for developing adaptive multimodal applications. In *IUI '06: Procs. of the 11th International Conf. on Intelligent User Interfaces*, Sydney, Australia, 2006. ACM Press.
- [Kaplan 05] Nancy Kaplan and Yoram Chisik. In the company of readers: the digital library book as "practiced place". In *JCDL '05: Procs. of the 5th ACM/IEEE-CS Joint Conf. on Digital Libraries*, pages 235–243, Denver, CO, USA, 2005. ACM Press.
- [Schilit 98] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. Beyond paper: supporting active reading with free form digital ink annotations. In *CHI '98: Procs. of the SIGCHI Conf. on Human factors in computing systems*, pages 249–256, Los Angeles, California, United States, 1998. ACM Press.
- [Schilit 99] Bill N. Schilit and Morgan N. Price *et al.* As we may read: The reading appliance revolution. *Computer*, 32(1):65–73, 1999.
- [vO01] Jacco van Ossenbruggen, Joost Geurts, Frank Cornelissen, Lloyd Rutledge, and Lynda Hardman. Towards second and third generation web-based multimedia. *The 10th International World Wide Web Conf.*, pages 479–488, May 1-5 2001.

Una Aproximación Sinestésica a la Interacción

Alberto José García Ariza
Grupo "Tracker . Laboratório Visual"
Facultade de Belas Artes.
Universidade de Vigo
albertoariza@uvigo.es

Sumario

En este trabajo se hace un repaso de la historia y evolución de las interfaces en la informática doméstica, a la vez, que se reflexiona sobre sus conceptos más importantes. Se propone un modelo sinestésico en la manera en la que operan los ordenadores, y se ejemplifica con pequeños experimentos realizados usando tecnología Flash.

Palabras-clave

Arte Digital, Historia, Informática, Interactividad, Interfaz, Sonido,

1. INTRODUCCIÓN

En los ordenadores, al igual que ocurre en la gran mayoría de las invenciones del hombre, siempre ha estado presente un aspecto muy importante: el de la comunicación o la operatividad del usuario con la máquina. Eso es lo que podemos definir como interacción.

Pero el problema nunca se encuentra en la presencia o ausencia de dicho elemento, sino en la calidad de esa relación. Y la calidad de esa relación es la que va a determinar que la usabilidad de dicho elemento se vea más o menos reforzada.

Evidentemente el estudio de esta relación entre hombre-máquina, o usuario-tecnología, se da en un ámbito borroso donde muchas competencias siempre tienen algo que aportar como psicología de la percepción, ergonomía o diseño.

Si restringimos un poco los conceptos sumamente amplios de los dos agentes que intervienen en el proceso interactivo; si pasamos de la tecnología, al concepto de ordenador; nos obliga a acercarnos al término de hombre a lo que conocemos comúnmente como usuario.

El objetivo del trabajo es precisamente este enfoque. El hacer un análisis de los medios tradicionales de los mecanismos de comunicación del usuario con el ordenador, observándolos desde una perspectiva de los servicios o funciones que los ordenadores nos aportan a nosotros; a la vez que mostrar tres pequeños ejemplos/experimentos que sirvan de explicación de diferentes aspectos relativos al tema de las interfaces.

Para ello vamos a usar en nuestros ejemplos el sonido como elemento no usado habitualmente, pero que se adapta perfectamente al concepto tradicional de interfaz.

2. CONCEPTOS CLAVE

Es preciso antes de empezar a explorar las lecturas que nos ofrece el uso del sonido como interfaz, el delimitar o acotar ciertos términos que usamos como herramientas metodológicas.

No es casual la inclusión del término sinestesia en el título de este trabajo. La sinestesia es la cualidad (me resisto a calificarla como enfermedad, y definitivamente no debe de ser considerada como una disfuncionalidad) de ciertas personas a entremezclar las sensaciones percibidas por sus sentidos. Es decir, un sinestésico es capaz de ver sonidos u oler colores, por ejemplo). De hecho, en su sentido etimológico nos remite a la unión de los sentidos, como si metafóricamente visualizásemos cada sentido en un hilo por separado que se unen en un nodo central, con lo que la relación entrada-salida no sea evidente.

Desde cierto punto de vista, no es gratuito el usar esta transmigración de los sentidos para entender el funcionamiento de los ordenadores. Precisamente, la interfaz lo que hace es establecer el camino de entrada de nuestras peticiones como usuarios, para que éstas sean operadas de la única manera que los ordenadores son capaces, esto es, matemáticamente

Por otro lado, es importante aclarar que las interfaces no son solo un mecanismo de entrada de datos, sino que también forman parte de la interfaz todos los dispositivos de output que muestran al usuario los outputs de los procesos informáticos. Al final todo se reduce a cuestiones de comunicación, y a diseñar los canales para hacer más fácil e intuitivo ese diálogo entre usuario-ordenador.

Al principio fue la línea de comandos. Así lo define Neal Stephenson, en su libro sobre la historia de la informática. Quizás no sea del todo cierto, ya que previo a ese estado,

los ordenadores se operaban con tarjetas perforadas, pero si nos centramos en la informática personal, podemos hablar que todo empieza en los sistemas operativos basados en un modelo de consola.

Este modelo carecía de lo que hoy conocemos como metáfora. En aquellos equipos, tanto los profesionales, como aquellos más domésticos, orientados al público infantil (Spectrum, Amstrad, MSX, Commodore 64), aún cuando disponían de ciertas capacidades gráficas, toda comunicación entre el usuario y el ordenador, debía hacerse exclusivamente mediante el teclado, ya fuera tecleando órdenes de programación o procesos, o accediendo mediante menús simples incluidos en el software.

Debido a que gran parte de los fines para los que se usaban aquellas máquinas eran los juegos, hubo gran difusión de dispositivos de interfaz más usables, pero exclusivamente de un tipo: Joystick (palanca de mandos), capacitando a los ordenadores de la usabilidad de las consolas de video-juegos. De hecho su propio nombre lo indica, joy-stick significa “vara de disfrutar”, explicitando el carácter lúdico de su naturaleza.

Durante esa época los ordenadores “profesionales” o “serios”, estaban destinados únicamente a personal especializado y entrenado en su uso. Pero a medida que la tecnología fue avanzando y difundiéndose, el ámbito de los ordenadores profesionales llegó a extenderse hasta espacios más domésticos, donde los usuarios no disponían de las habilidades técnicas para poder operar con ellos de manera eficiente.

Empezaron a surgir aplicaciones de software más universales (esto es que servían para hacer tareas que cualquiera podía necesitar) como procesadores de textos, sistemas de contabilidad, hojas de cálculo. En esta época emergieron los sistemas operativos, que ejercían de puente entre el usuario y la máquina, ejecutando todas las tareas administrativas para manejar el sistema. Seguían basándose en la línea de comandos pero a un nivel más alto, es decir, más cercanos a la lógica del usuario, que a los procesos de computación internos del ordenador.

Por aquella época, en el ARPA (Advanced Research Projects Agency) estaban investigando como poder superar la línea de comandos, y desarrollaron dos ideas que conjuntamente han cambiando el uso de los ordenadores. El ratón y la metáfora.

La idea de la metáfora buscaba solucionar el problema de cómo presentar en la pantalla del ordenador toda la información dinámica que todo usuario necesita disponer. La innovación vino, una vez más, al fijarse en el campo de los videojuegos. Al igual que cuando se maneja un juego electrónico, nos olvidamos del programa en sí, y nos sumergimos en la historia y espacio que el videojuego nos presenta, se propuso el presentar la información de una manera más tangible, como si fueran objetos del mundo real.

Debido al uso empresarial para el que estaban destinados los ordenadores, la metáfora que se escogió para representar

la información y tareas que los ordenador podían hacer fue precisamente esa, la de la oficina. De esa manera surgieron los iconos, y el espacio a través del cual nos movemos, usando un dispositivo nuevo, capaz de desplazarnos por dicho espacio, el ratón, cuya encarnación en la pantalla, suponía el desdoblamiento del usuario en puntero, casi como desarrollo evolutivo del antiguo cursor.

Esta metáfora fue ampliamente difundida por Apple, en su ordenador Lisa, a principios de los 80, y luego rápidamente extendida al resto de sistemas operativos de escritorio. Lo que demostró Apple, con Lisa, fue que con la metáfora del escritorio, la papelera, los iconos de archivos y carpetas; que realmente no había que conocer como funciona un ordenador para poder operar con él.

De ese momento a hoy en día, las aplicaciones informáticas se han vuelto más complejas y sofisticadas, así como sus interfaces, pero en esencia siguen siendo las mismas. Podemos tener ventanas con bordes y texturas más pulidos, iconos que proyecten sombras, distintos escritorios virtuales, pero en esencia poco se ha evolucionado hacia una nueva revisión del concepto de interactividad con el usuario, quizás en la medida de que no se ha encontrado una manera de superar al ratón con dispositivo físico de interacción más operativo. A principios de los 90 hubo un cierto boom de la incipiente tecnología de la realidad virtual, pero nunca fue demasiado operativo debido a básicamente a dos razones. Por un lado, no era muy real, debido a la potencia necesaria de los ordenadores para poder manejar los entornos virtuales; y por otro, que no es demasiado cómodo el tener que enfundarse en todo tipo de trajes, guantes, sensores, cascos o gafas simplemente para poder realizar tareas básicas como chequear un correo electrónico.

Hoy en día, podemos percibir una cierta tendencia en la informática personal. Se tiende a la unificación de servicios. Esto es, que disponemos en nuestra vida cotidiana con una multitud de aparatos electrónicos que nos permiten básicamente poder realizar las mismas funciones desde distintas ubicaciones y dispositivos. Pensemos en las PDAs, tablet PCs, teléfonos móviles, consolas de videojuegos portátiles con conexión a Internet, reproductores de MP3, GPS.... En definitiva, se tiene que tender a la portabilidad y a la unidad. Y se busca dentro de eso una metainterfaz, un dispositivo que sirva para todas las funciones particulares mencionadas, a la vez, que sirva para conectarse y que pueda ser usado como un ordenador normal. Varios ejemplos de hacia dónde puede estar investigándose en ese aspecto de construir la metaherramienta puede verse en los “fakes” de Internet sobre los ficticios productos Ipod Touchable de Apple, o el proyecto Origami de Microsoft.

Por otro lado, en el sector lúdico de los videojuegos vemos como se están difundiendo nuevas maneras de interactuar con el software. Las consolas de videojuegos no están innovando tanto en la conectividad de elementos

nuevos, sino en la manera en que esos elementos pueden ser usados como medios de operatividad con el software.

Hoy en día el sector de los videojuegos cuenta con gran variedad de interfaces no convencionales, entendiendo por no convencionales, todas aquellas que se alejen de la idea del joystick o del pad de cursores. Desde hacia varios años podemos jugar a videojuegos de snowboard o skateboard, subidos a una tabla o conducir cualquier tipo de vehículos en réplicas de los mismos, por ejemplo. Todo este tipo de interfaces, se basan en la mera réplica de un objeto del mundo real, que operado de la misma manera, funciona como una interfaz de una simulación de dicho objeto en el mundo electrónico del ordenador. Este modelo de interfaz, más física, que imita al mundo real, también se ha introducido en el mundo de la productividad y en el ámbito doméstico. Basta con pensar en las tabletas gráficas, que permiten “pintar” digitalmente, y vemos como la metáfora se extiende también al software específico. como Adobe Photoshop, que toda su metáfora se basa en la iconografía de los útiles de pintura y fotografía.

Los interfaces anteriores nos hablaban de simulación, innovación en la forma pero no el medio de interactividad. Por otro lado, podemos distinguir otro tipo de interfaz donde se usan elementos físicos más vacíos de significado para poder construir alternativas de interfaz distintas. Cuando me refiero a si un elemento está más o menos vacío de significado, me refiero que por ejemplo, una guitarra sirve para tocar música y se usa de una determinada manera muy característica. El objeto en sí tiene un gran bagaje y peso sobre luego las aplicaciones para las que se pueda pensar. Lo normal es que se use como interfaz para un software de simulación musical. En cambio, si pensamos en un objeto con poco “peso” de significado, como una cámara, o un micrófono, cuya función es simplemente el capturar ciertos inputs, su uso puede ser mucho más versátil.

En estos últimos 3 años, por un lado el abaratamiento y expansión de las cámaras digitales, ha permitido que se use las webcams como interfaces de aplicaciones con distintos usos. Desde medios de navegación en espacios (sustituyendo a la cara tecnología de sensores, mediante software de reconocimiento), interacción con formas virtuales, juegos de coordinación corporal, y entrenamiento deportivo.

Por otro lado, la expansión tanto de tecnologías dinámicas en la web, como la expansión de posibilidades de comunicación mediante voz, ha hecho que desde el arte y la experimentación plástica se “perviertan” las posibilidades de operar mediante el micrófono, para ofrecer nuevas experiencias a un medio, en principio tan cerrado como es la web.

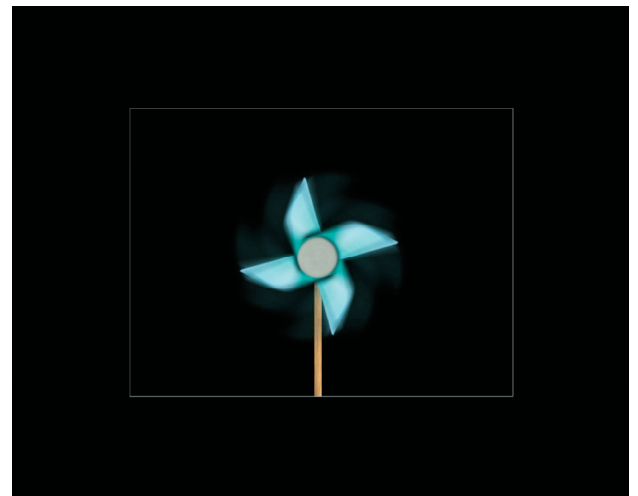
En ese sentido hay que señalar trabajos, no excesivamente complejos técnicamente a un nivel de tener que ser realizados en un laboratorio, pero que sí son totalmente innovadores por los procesos de inmersión en los espacios digitales que nos proponen. Bolter y

Gromala, nos proponen en su libro “Windows and Mirrors”, una perspectiva en la que los ordenadores son a la vez opacos y reflectantes. Opacos en el momento que no separan de lo que realmente están haciendo, atendiéndonos como sirvientes que se preocupan de que tengamos lo que queremos, y que a la vez no nos tengamos que preocupar de cómo se consiguen nuestros deseos. Pero esta opacidad, a su vez, se desdobra en un reflejo. Los ordenadores, hacen que su fachada sea un reflejo cada vez más exacto del mundo que tenemos fuera de las pantallas. Se tiende a que cada vez sean más invisibles, ya que no dejan ver que están ahí, y en su lugar sólo percibimos más mundo real.

El potencial de este tipo de trabajos reside en su carácter sinestésico. Es cómo son capaces de aprovechar la capacidad del ordenador de convertir todo input en un denominador común matemático, que permite su traducción a cualquier otro output. De esa manera podemos pintar con nuestra voz, con la *Messa di Voce* de Golan Levin; o podemos componer y música espacialmente, con los ejemplos de metáforas de sonido binaural de Santiago Ortiz, por poner sólo un par de casos.

En este sentido, presento dos experimentos sencillos, que ilustran dos casos clave, la simulación como tal, y la simulación como metáfora.

3. ORIGAMI



La obra se compone básicamente de una aplicación software programada para que reaccione a los estímulos del usuario, mediante una interfaz no convencional. Es un molinillo de papel virtual, al que se le han aplicado los comportamientos físicos y cinéticos para que responda al estímulo del sonido que le llegue mediante el micrófono, de manera que cuando el usuario sopla a un micrófono situado enfrente del monitor, el molino reaccione al soplo como lo haría un molinillo de papel convencional.

No es un experimento nada complejo ni a nivel conceptual ni a nivel técnico. Simplemente se trata de explorar los conceptos de simulación, de interfaz, y las

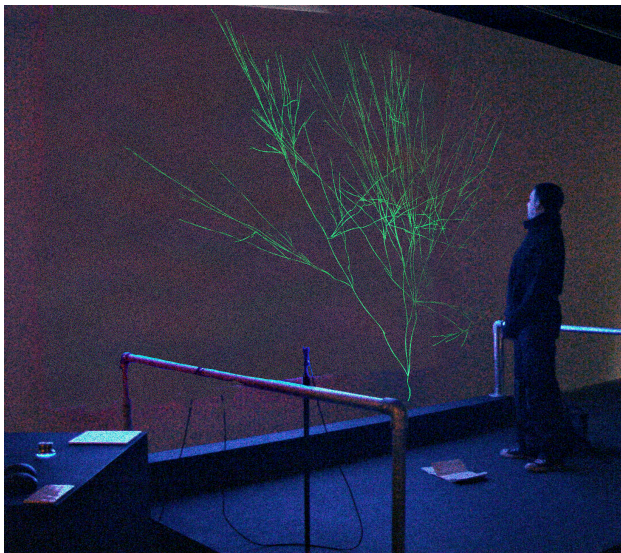
capacidades sinestésicas de la creación digital en tanto que todo input se procesa a nivel matemático, del que se destila un output o resultado

4. FONOSINTESIS

Fonosíntesis es una instalación audiovisual en la que se representa un organismo de morfología inspirada en lo vegetal, y cuyo mecanismo trófico se basa en el sonido, entendiendo esos sonidos como materia prima desde una perspectiva formal y fonética.

La instalación trata temas como la comunicación abstracta, las relaciones sinestésicas entre sonido y forma, y todo ello presentado dentro de un sistema audiovisual lúdico, enmarcado en un entorno de complejidad.

Surge de la necesidad de hacer transitable la comunicación-acción-reacción-interacción entre el espectador y la obra interactiva. Trata de romper las limitaciones tácitas existentes entre el software y el usuario, que tienden a constreñir el input a la operatividad de un ratón y un teclado. Por ello se propone no solo la modificación de las formas generadas mediante el sonido, sino también el control de ellas por parte del usuario.



Fonosíntesis es un prototipo que reflexiona sobre la creación digital como ser vivo, como organismo autónomo con sus propias características y comportamientos perfectamente definidos. Habla de la unión de conceptos y mundos virtualmente aislados. Las obras digitales están separadas del mundo real en el que se encuentra el espectador. Es fundamental el dejar eso claro de manera implícita en la propia pieza.

El concepto de fonosíntesis nace del juego de palabras y de la proximidad fonética al vocablo fotosíntesis, que se refiere procesos metabólicos de los que se valen las células para obtener energía. Ya en la propia definición del término se presenta una característica fundamental que está presente en toda obra digital: la transformación o traducción de un tipo de elemento en otro. De luz en vida.

El comportamiento se define de la siguiente manera. En todo momento hay n ramas en proceso de crecimiento. Ya sea el primer caso, donde solo habrá el tronco principal, o cualquier otro momento donde habrá más ramas. Las ramas que tienen hijos ya han completado su proceso de crecimiento y en principio están apartadas del modelo.

Entonces, cualquier rama que esté en proceso de crecimiento (esto es, que no haya crecido del todo y por tanto no tenga hijos) necesita del sonido del ambiente para desarrollarse. En cuanto hay un sonido en el ambiente ese sonido es traducido a número y transportado a una rama como energía que será usada para desarrollarse.

El paquete de energía que recibe una rama no es usado de manera instantánea, sino que incrementa las reservas de energía de la rama, lo que le permite hacer un crecimiento dilatado en el tiempo.

Cuando una rama llega a su desarrollo máximo se produce una bifurcación. Es decir, su proceso de reproducción decide cuantos hijos va a tener en función de la reserva de energía que tenga en ese momento, y a partir de ahí cede la energía para que sus hijos empiecen a desarrollarse, y deja de estar pendiente del sonido, dejando el lugar a sus hijos.

Cuando nace una rama, lo primero que hace es escoger su morfología aleatoriamente de un banco de ramas, la diversidad proporciona una cierta ilusión de realidad. Una vez decidido su aspecto básico, termina de configurarse mediante ciertos aspectos heredados de su padre, pero con ciertas peculiaridades que le hacen única. Cada rama preserva la inclinación de la rama de su padre, pero hay un cierto rango de aleatoriedad necesaria para que la representación no sea perfectamente geométrica. También toda rama conserva el grosor de su padre, pero a su vez, va disminuyendo el grosor propio en cierto porcentaje, y que a su vez transmitirá a sus hijos. Básicamente ese es el comportamiento del sistema desde la perspectiva de la planta, pero si lo observamos desde el punto de vista del sonido se pueden apreciar nuevas particularidades.

El sonido registrado en todo momento es recogido por un micrófono, y el software lo codifica a niveles de intensidad, con valores que van de 0 (silencio absoluto) a 100 (ruido total). Cada energía se usará en el crecimiento de una rama. Se podría establecer el símil de que son paquetes de comida, y un mismo alimento no puede ser ingerido por dos organismos distintos. El mecanismo que se usa entonces es que hay una lista de ramas candidatas a recibir esa energía, es decir ramas en proceso de crecimiento, y un puntero se ocupa de ir desplazándose por la lista. Es decir, situando en estado de recepción a las ramas de una en una, y una sola en cada momento, con lo que se produce una asignación directa del sonido generado en cada momento a una sola rama. Cuando una rama ha completado su desarrollo, y se ha bifurcado, se borra de la lista y se añaden a la lista todos los hijos que haya generado.

Como se ve, el comportamiento de la planta se define en términos narrativos, y usando como metodología un proceso de síntesis, que no tiene por que ser completamente real o vigente en el mundo real, pero si que ha de ser completo y coherente. Una vez definido de manera total dicho comportamiento, teniendo en cuenta el funcionamiento global y local del sistema es cuando se puede producir a la codificación de dichos patrones y a su programación.

5. AVION

Un tercer ejemplo, a modo de conclusión, nos muestra nuevas posibilidades del uso del sonido como input, pero aplicado a modelos de interfaz totalmente convencionales.

Nos situamos ante un escenario clásico de videojuego. Disponemos de un avión que debe evitar cierto tipo de obstáculos, con la única diferencia que en el juego tradicional normalmente manejaríamos el avión mediante el teclado o el ratón, mientras que aquí el avión es

manejado mediante el uso del sonido. A más sonido, mayor energía le transmitimos al avión, y éste sube. En ausencia de sonido, el avión usa su energía residual y desciende.

Este modelo es perfectamente cerrado, ya que para el funcionamiento de este software, solo precisamos de un control (encendido-apagado), con lo que es perfectamente compatible con el binomio (sonido-silencio). En este caso, además, la intensidad de sonido tiene un rango de 0 a 100, con lo que permite una mayor flexibilidad y riqueza en la aplicación.

6. REFERENCIAS

- Bolter, J.D. y Gromala, D.R. Windows and Mirrors. Interaction Design Digital Art and the Myth of Transparency. (MIT Press, 2003).
- Stephenson, N., In the Beginning was the Command Line (1999)
<http://www.cryptonomicon/beginning.html>
- Wooley, B. El universo virtual. (Acento, 1994).

Gesture Human-Computer Interface for Command and Control

José Miguel Sales Dias (*,**)
midias@microsoft.com

Nuno Barata (**)
Nuno.Barata@iscte.pt

Pedro Nande (**)
Pedro.Nande@iscte.pt

André Correia (**)
Andre.Correia@iscte.pt

(*) MLDC, Microsoft Language Development Center, MSFT LDA, Av. Prof. Aníbal Cavaco Silva, Edifício C1-C2, Tagus Park, Porto Salvo, Portugal, www.microsoft.com/portugal/mldc
(**) ADETTI/ISCTE, Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática, Edifício ISCTE, 1600-082 Lisboa, Portugal, www.adetti.pt

ABSTRACT

In this paper, we describe, test and validate through usability evaluation, a hand gesture recognition engine based on computer-vision. In our approach, we use a computer, with limited resources and a simple video camera, as a basic platform to enable the development of generic gesture-based Human-Computer Interfaces appropriate for Command and Control (C&C) tasks in computer applications and at interactive rates. With our technique, the system removes initially the background of captured images so that irrelevant pixel information is not considered in further analysis. The human hand motion is then detected, segmented and its contours localized, to allow a comparison with a pre-defined static hand poses library using various algorithmic approaches, some image-based other contour-based. Each hand pose, once recognised, can activate Human-Computer Interface command and control actions, organised in a hierarchy manner, which are specific for a given application. This work has been concluded in close collaboration with the Portuguese Foundation of the Communications, an institution owned by Portugal Telecom, the largest Portuguese operator. This cooperation has enabled the development and deployment of a demonstration application, using this platform, in the “House of the Future” test-bed, of the Communications Museum in Lisbon, Portugal. In the application, we enabled the command and control of some consumer devices linked within a home network, such as controlling the TV, the room lighting or the windows opacity characteristic. The paper also presents experimental results, regarding the precision and usability of the implemented system and discusses the best algorithmic approach for the purpose of identifying static hand poses at almost real-time rates and analyses future directions of our work.

Keywords

Gesture Recognition, Hand Poses, Computer Vision, Human-Computer Interaction, Command and Control, Portuguese Sign Language, Home Network

1. INTRODUCTION

The recent development of applications in the domain of computer vision has enabled the exploration of new approaches in the way humans interact with machines and with computers. The ideal form of interaction would be, for a given person, to have a familiar way of accessing specific computing application functionalities, without the need of using specific peripheral devices that could distract the user from the focus of his/her task. That could mean for the same person, to communicate with a machine much in the same way he/she would interact with other humans. There has been a considerable evolution in the issue of natural modalities of human-machine interaction, and new solutions have shown relevant

improvements and have demonstrated that it is possible to use different modalities, such as speech, gesture or other natural media as multiple ways to communicate with the computer. With our work we are aiming at developing an open source gestures recognition engine that would offer the possibility to trigger user-specified command and control actions, activated by different hand gestures. The system requires a single video camera linked to a computer that captures the user's movements and recognizes hand gesturing, namely, a set of pre-defined static hand poses. A “static hand pose” is a gesture represented by a single hand pose with a spatial position that doesn't vary much in time. As an example, it could be a symbol of a sign language alphabet [Kadous95] [Niwa02]. This type of gestures, which are simple to

issue by humans, can trigger actions in the application level providing human-machine interaction. We have followed this approach in the “House of the Future” test-bed of the Communications Museum in Lisbon, Portugal, where we have developed a demonstration application that uses gesture-based HCI in a living room scenario. In our set-up the application has access to the home network of the house, which enables the user to interact and control some of the home networking linked devices like, a TV Set-Top-Box, the window glass transparency characteristics (opaque/transparent) and activating macro commands, such as opening/shutting the lights and opening/shutting down the blinds at nightfall. Our analysis of the Portuguese Sign Language signs [Hub98] has largely influenced our selection of the static hand poses, whose detection and identification as triggered our technique of gesture-based HCI. In synthesis, our system recognizes a set of static hand gestures based on computer vision that are used in an application that interprets those gestures and activates hierarchical actions, like controlling home devices and changing indoor’s ambient conditions. With our technique we are able to achieve interactive rates (near real time performance, with an average of 6 frames per second on a Pentium IV with 3 GHz CPU, 512 MByte RAM, NVIDIA GeForce 4 with 64 MByte and 640x480 video input resolution). The paper is organized as follows: in section 2. RELATED WORK, we provide a short background in the issues of gesture data acquisition and recognition. In section 3. OPEN GESTURES RECOGNITION SYSTEM ARCHITECTURE, our system architecture, engine modules and dataflow, including the way we classify the type of recognised gestures and triggered actions, supported by our system. In section 4. PRECISION TEST RESULTS AND DISCUSSION, we detail our experimental performance testing results and discuss about the best studied methods for robust static hand shape recognition. In section 5. APPLICATION OF GESTURE HCI FOR COMMAND AND CONTROL, we describe the development, user interface, available functionalities and deployment of a demonstrator, which has been set in the “House of the Future” of the Communications Museum in Lisbon, Portugal. By using this test-bed we detail, in section 6. USABILITY EVALUATION, the usability evaluation results obtained with the mentioned application in the “House of the Future”. Finally, in section 7. CONCLUSIONS AND FUTURE RESULTS, we derive conclusions of the research done and point out future directions for our work.

2. RELATED WORK

In this section we will present a short overview of related work in the domain of gesture recognition. There are two different approaches when acquiring data for gesture recognition: one based on sensory devices and the other a computer vision approach. There are also hybrid solutions that combine both fields, aiming towards a pragmatic solution.

2.1 Approach Based on Sensor Devices

This approach is focused in Mechanical, Electronics and Electromagnetic Engineering research and suggests the use of physical devices that can measure the variations of certain values that occur while performing a gesture, such as position changing, orientation, accelerations and forces made by the hand. The gesture analysis is essentially mechanical and electromagnetic, consisting in the evaluation of these physical parameters. Due to the specificity, complexity and cost associated to this type of electronic devices the systems based on this approach are usually less scalable and expensive. Nevertheless, there are academic developments of these solutions [Kadous05] [Geoffrey98] and commercial applications with some acceptance in the market.

2.2 Approach Based on Computer Vision

Gestures recognition systems based on computer vision are subject of widely diffused investigation works. Although theoretic approaches tend to perspective user gestures in the large and general problem of human motion, several results have focused on practicality and usability. Therefore, there are gesture-recognition specific systems of great interest. In [Chaudhury00] [Niwa02] [Shirai02], we find gesture recognition systems applied to Sign Language interpretation. Other systems, like in [Aguilar03] are more directed for human-computer interface control. Recognition techniques, either image or vector based, all tend to acquire hand contour as a starting point for gesture perception and then its motion, have been somewhat influenced by human visual system processes. Background complexity is not frequently addressed, requiring gestures to be executed against a homogenous prepared scene. Related works from the computer vision domain, like advanced background subtraction [Davis99], can be useful for gesture recognition. Alternative approaches are based on mathematical descriptions of the whole captured image itself, as in [Maydt02], where Haar-like features are used to recognize trained objects, with an interesting potential for hand detection.

3. OPEN GESTURES RECOGNITION SYSTEM ARCHITECTURE

3.1 Architecture Overview

Our O.G.R.E, Open Gestures Recognition Engine (not to be confused with a popular 3D Engine [OGRE3D]), simplified modular system architecture and process flow, is depicted in Figure 1.

The system is conceptually divided in 3 abstract layers, from bottom (level 1) to top (level 3), which were defined in order to tackle the problem of defining a gesture recognition engine in a generic way:

- Level 1: Preparation, extraction and manipulation of visual information from a sequence of images.
- Level 2: Low level gesture recognition;
- Level 3: Interaction with the application layer.

In level 1, we have Video Capturing, responsible for image acquisition from the video camera, which feed the

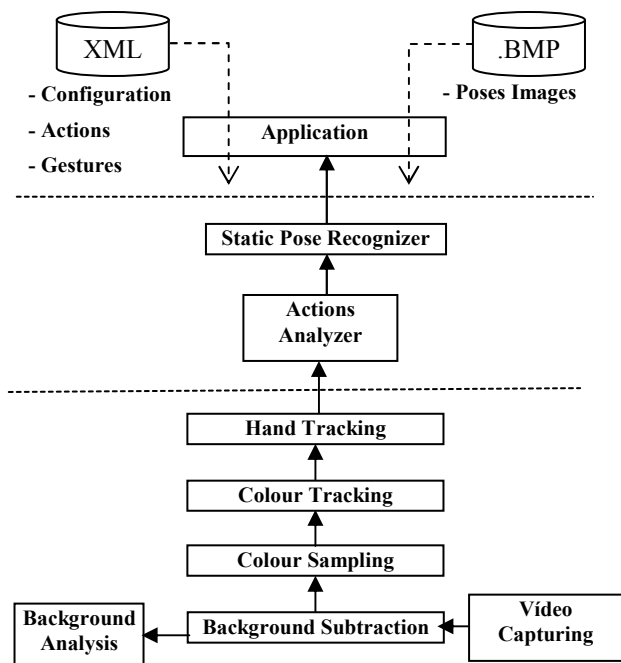


Figure 1. O.G.R.E. System Architecture.

Background Subtraction module. This last one provides the adaptive elimination of the static background, being aided in this task, by the Background Analysis Module. Once processed, images are sent to the Colour Sampling module, where the human skin tonality can be sampled or parameterized. With this information, the colour tracking module, scans subsequent images, looking for the skin Hue tone. With the found tonality area, the Hand Extraction module, will obtain the hand contour and pixmap mask, which will be optionally processed by the Contour Handling and B-Spline modules, to filter any contour noise, through mathematical approximation techniques. In Level 2, the parameters obtained in the previous Layer, are processed by the Actions Analyser Module, responsible for the XML configuration of the different gesture recognition algorithms and the invocation of the corresponding algorithm (only Static Pose Recognition is used in this paper, but the system also supports Simple Path Recognition, Staged Path Recognition and Free Path Recognition). Once a specific gesture is recognised and the corresponding Action is identified, by this Layer, the result is communicated to the Application Layer. Level 3 is composed by the Application module, which represents then user application that creates and invokes the OGRE engine. The XML Gesture-Action configuration file and the pixmaps with the known 2D static hand poses are supplied by the application to the engine.

3.2. Gestures and Actions Definitions

Static Poses

Static poses are rigid hand postures (a type of gesture according to our classification) which do not depend on the movement of the hand. They are trivially characterized only by their shape in the form of a 2D silhouette (a bitmap) stored in a user-specific database.

A static pose is defined in XML by the following notation:

```
<StaticPose name="A" filename="A.bmp">
</StaticPose>
```

Actions

Our engine introduces the abstract notion of actions, which are contextualized hierarchies of application dependent functionalities (requiring static hand gesture recognition for HCI), which feed the gestures engine and provide a guide for available recognition algorithms selection. Actions inform the application of the type of gesture to recognize at a given moment, thus minimising the need to activate all the different gesture recognition types simultaneously. Actions also simplify the communication between the engine and the application, since this last one does not need to know what gestures have been recognised, but only which actions have been activated. An Action corresponds to an application functionality. It is compose by an activation gesture and an optional termination gesture and, if required, by a list of optional child actions, which may be activated on a given application context.

There is an XML description of gestures and actions, which is useful for configuration purposes. A simple action is composed by a valid starting and an ending gesture (this last one is optional), which, respectively, trigger and terminate the action. In the context of this paper, a valid gesture is a StaticPose:

```
<Action name="mySimpleAction1">
  <Start gesture_name="A"></Start>
  <End gesture_name="B"></End>
</Action>
```

An action can also be composed of child actions previously defined, accessible in its context. These can be simple or composed actions:

```
<Action name="myComposedAction1">
  <Start gesture_name="C"></Start>
  <End gesture_name="D"></End>
  <Actions>
    <Action name="mySimpleAction1"></Action>
    <Action name="mySimpleAction2"></Action>
  </Actions>
</Action>
```

There is a Root Action that determines the topmost hierarchical level and those identified as its children corresponds to the first hierarchical level of known actions:

```
<Action name="root">
  <Start gesture_name=""></Start>
  <End gesture_name=""></End>
  <Actions>
    <Action name="myComposedAction1"></Action>
    <Action name="mySimpleAction1"></Action>
    <Action name="mySimpleAction2"></Action>
  </Actions>
</Action>
```

For the purpose of the application installed in the “House of the Future”, we have identified one type of hand gestures, suitable for simple forward recognition: Static

Hand Poses. However, in addition to Static Hand Poses, the O.G.R.E engine also supports Simple Paths (primitive shapes such as circles, squares and triangles), Staged Paths (a mixture of Static Hand Poses and Simple Paths) and Free Tracking [Dias04], which are out of the scope of this paper.

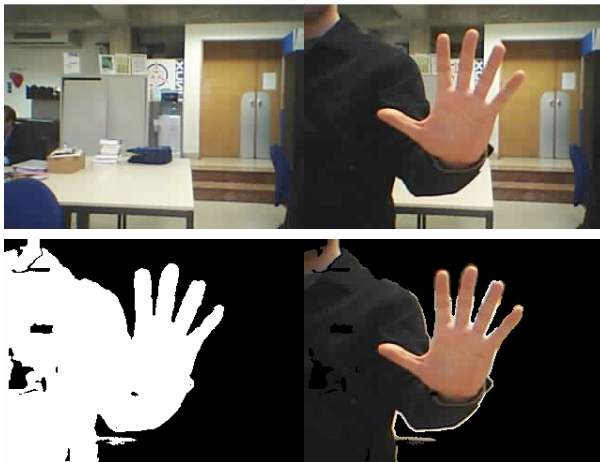


Figure 2. Subtracting the background. Top Left: the original background. Top Right: The moving user in foreground. Bottom Left: the mask with calibrated background removed and subsequent changes maintained. Bottom Right: Foreground user with subtracted background.

3.3. Engine Modules

Background Subtraction

Background subtraction [Davis99] is applied prior to any subsequent processing. It consists of a calibration period during which maximum and minimum per-pixel values in the YCrCb domain are stored and updated. After this initial period, foreground classification occurs, based on simple comparison between actual frame pixels YCrCb values and the stored background model, since it is assumed that variations of actual frame pixels YCrCb values below the stored minimum or above the store maximum, classify them as foreground pixels (Figure 2).

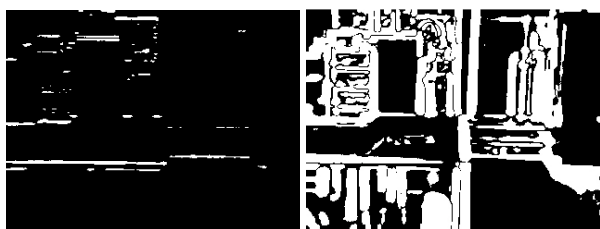


Figure 3. Background subtraction mask noise due to lighting changes (left) and camera positioning variation (right).

Background Analysis

This module is responsible for background deterioration detection (Figure 3). It has been observed that the background subtraction algorithm used is not resilient to environmental changes, such as light fading, scene decorative objects replacement and camera positioning instability. Therefore, for long functioning periods, it is necessary to robustly adapt to new scene conditions without user intervention. The algorithm used for

detecting such changes is based on a timed observation of the background subtraction mask. In normal conditions, this mask is a binary image composed of both black and white pixels, representing background and foreground regions respectively. Foreground regions, as typically observed during user interaction, are a reduced number of large connected white areas (about one or two spots of a quarter image area). When deteriorated, this mask will contain several smaller regions spread throughout expected locations (the edges of contrasting background elements, such as closets, doors, tables, etc). These spots can therefore be classified as noise, as they are unwanted, disruptive elements for both background and colour segmentation. Their detection is done by finding small separated regions of ‘white’ value. For noise classification purposes, two measures are considered: 1) a maximum fraction of occupied area of all spots, relatively to the image (a configurable parameter), above which it is seen as noisy, and 2) the minimum noise coverage area, relative to the image’s area, above which it is considered large enough to cause interference (also, a configurable parameter). A second type of noise is also observed when environmental conditions vary drastically (turning lights on or off, camera’s field of view occlusion by large passing objects, etc). In all these cases there are no small spots visible, as the entire mask is constituted by a large white stain. This is also disruptive and therefore, considered noise. Its detection consists on simply establishing a maximum area for white value regions occupation (a configurable parameter), above which such elements are taken in consideration as noise. In both algorithms, noise detection triggers a positive alarm. During normal interaction a time window of such alarms is analyzed. If these positive alarms appear in a considerable number, recalibration will occur automatically (Figure 4). During calibration, no time window is analyzed, as a unique positive alarm is sufficient to destroy the entire statistical model

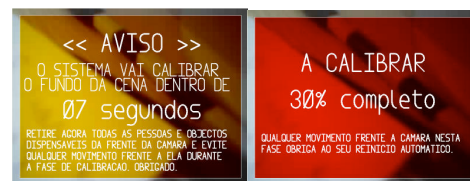


Figure 4. Automatic re-calibration of the scene: Left – Notification of the re-calibration; Right – re-calibration process.

Colour Sampling

In each frame, the hand is localized via colour information based on a Hue interval, either user specified by a specific initial interaction (the user approximates is hand to the video camera) or dynamically sampled, as in [Niwa02] or [Shirai02]. Our technique has proven to be experimentally robust to luminance variation and easily finds the skin tonality in a given environment.

Colour Tracking

Hue values obtained in the previous colour sampling phase, feed the CAMSHIFT algorithm (Continuously Adaptive Mean Shift [Bradski98]), which is then applied

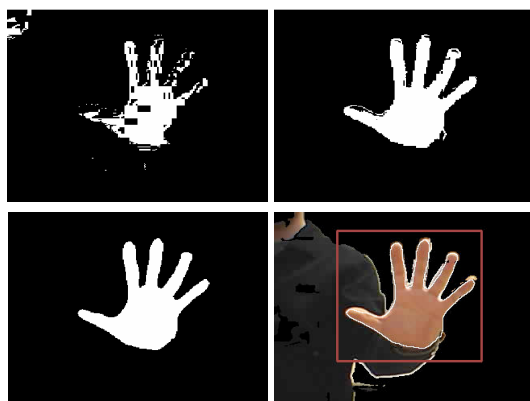


Figure 5. Extracting hand contour. Top left, the CAMSHIFT algorithm histogram back projection result. Top right: Result after YCrCb re-sampling. Bottom Left: result after morphological smoothing. Bottom Right: result after contour vectorization (white contour), with CAMSHIFT window limited to the green bounding box and mixture with foreground image.

to the current captured image. The CAMSHIFT algorithm computes a histogram back projection binary image (with reduced resolution, in respect to the initial image luminance, since the algorithm works in the sub-sampled chrominance domain), representing areas of a specified Hue tonality (the hand tonality, in our case). It also reduces the hand pose searching area to the largest connected component representing the user's hand Hue. Hand tracking is therefore guaranteed in the following frames (Figure 5).

Hand Extraction

The resulting hand contour image is still inadequate for static pose recognition. In order to extract a realistic hand silhouette, further processing is needed. In our algorithm, the histogram back projection binary image is used as a mask for YCrCb color space re-sampling, applied to the initial image with the background removed. With this, we intent to obtain richer information from image planes more suitable for edge extraction. Luminance and chrominances are sampled at different resolutions in order to achieve the best possible contour detail. With this process we are able to determine the average of YCrCb pixel values of the hand region, and perform a better classification of pixels, belonging or not, to the hand contour and interior. A smoothing morphological operation is then applied with an adequate structural filtering element for noise reduction. This element has dimensions which can be of 5x5, 7x7, 9x9 or 11x11 pixel, depending in the estimated silhouette dimension. The contour is then vectorized and, if necessary, a polygonal approximation sensitive to finger curvature is applied. This approximation is based on the best fit ellipse mathematical approach, as to obtain a measure of a given set of point's curvature, proportional to the ellipse eccentricity.

Actions Analyser

This module is the engine's core "intelligence". It analyses a specific action context and redirects gesture recognition into the adequate set of possible hand poses

recognition algorithms (in the case of this paper, only static hand pose is considered).

Static Hand Pose Recogniser

In order to recognize static poses, several widely known algorithms for shape analysis were studied. We can divide these algorithms in two categories, Image Based Analysis and Contour Based Analysis. We have studied the following:

Image Based Analysis:

1. Template Matching [Bradski02]: Based in the convolution between two images at several scales in order to find a known template. It is scale invariant but variant to rotation and translation.
2. Discrete Cosine Transform Analysis: [Gianino84] A scale and rotation independent transformation in the frequency domain.

Contour Based Analysis:

3. Hu Moments [Bradski02]: These are a set of shape characteristics, invariant to rotation and scale metrics that can be useful for shape classification.
4. Pair-wise Geometrical Histogram (PGH) [Heuer84]: This method, computes the Histogram of distances and angles between the contour polygon's edges, which provides us with a unique contour signature. PGH is scale invariant and symmetry invariant as well (in relating to the xx' and yy' image axes). However, PGH is rotation variant.
5. Simple Shape Descriptors (SSD) [Lu02] [Fonseca00] [Eckhardt00] [Wirht02], combine simple metrics which help describing shapes.
6. PGH-SSD Hybrid: This method corresponds to the authors efforts in combining the PGH and SSD advantages.
7. CALI [Fonseca00]: This is a software package, based in Fuzzy Logic, used normally to recognise sketched shapes in the context of calligraphic interfaces. The technique may bring advantage in the recognition of static hand poses, by introducing a probabilistic methodology in the recognition technique. It is invariant to rigid body transformations (scale, rotation and translation).

For static Hand Pose Recognition, the extracted hand silhouette is compared against a library of silhouettes templates or a library of silhouettes signatures (depending in the method), using one of the above algorithms. All algorithms were tested regarding its efficiency and precision, as described in the next section.

4. PRECISION TESTS RESULTS AND DISCUSSION

As a test case, static hand pose recognition algorithms were tested for the Portuguese Sign Language (PSL) 36 signs (more precisely, for 2D configurations of such signs, see Figure 6) and an Average Recognition Rate (%) was recorded. The selection of this test case, had to do with the fact that it constitutes a complex problem with a variety of 2D hand shapes, not yet addressed in the literature, for the particular PSL case (although considerable work does exist for other Sign Languages).

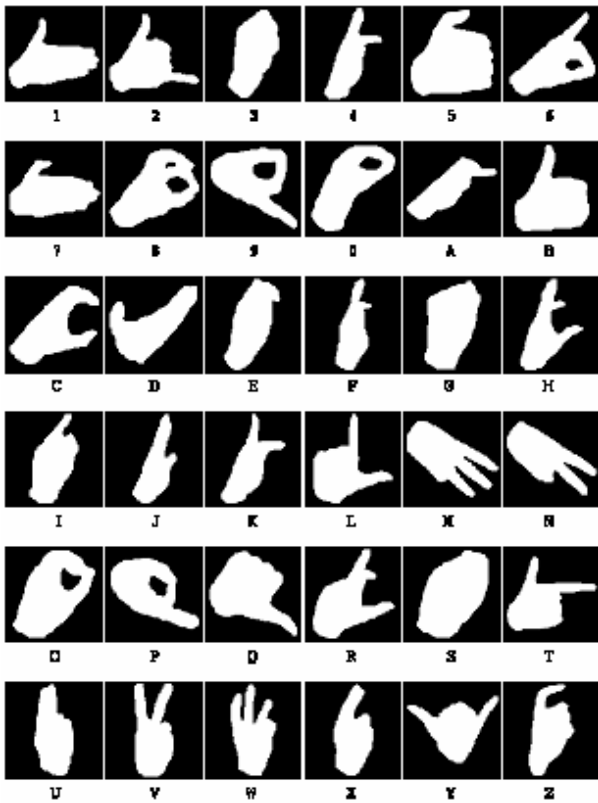


Figure 6. Portuguese Sign Language: 2D Sign configurations.

Processor	Clok Freq.	Graphics	Video Input Resolution ¹
Intel Pentium IV	3 GHz	NVIDIA GeForce 4	640 x 480

Table 1. Hardware configuration used for the precision tests.

The interest of the authors of pursuing, in the future, studies towards full PSL recognition, has influenced also the choice of this use case. These were the experimental conditions:

1. Each hand pose was tested ten times against a user library (that is, the subject performing the gestures was the same user whose hand poses had been stored in the library) and, against another person’s library (the subject performing the tests was not the same one, whose hand poses images were in the library). Taking into account the number of PSL symbols, of libraries (user and generic, except for the CALI technique, which just use a generic library) and of hand pose recognition algorithms (7), this resulted in 4680 static hand pose user trials: $4680 = (36 * 10 * 2 * 6) + (36 * 10)$
2. The scene background consisted in a heterogeneous environment, with various objects with different colour hues (some approaching the skin tonality).
3. Given the existence of various light sources in the environment, and in order to minimise zones of super-exposition of light in the user hand, an area of shadow occupying the hand was created, to

homogenize the luminous energy in the area of the hand.

4. The subject was wearing a long black shirt that effectively eliminated the harm skin, which interfered with the hand pose recognition algorithm.
5. The subject was knowledgeable of the PSL signs and the gesture technique to execute them, although he had no specific practice. With this option, the idea was to emulate an average user that has never practiced PSL and that could introduce human-originated errors in the process of generating the signs.
6. A given sign was considered to be recognised with success, if the system was able to identify it correctly almost immediately and, to stabilise the recognition during some seconds

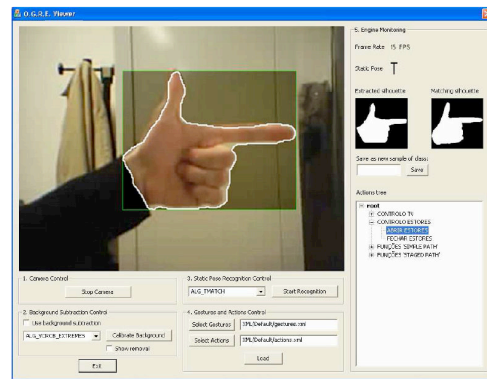


Figure 7. User Interface of O.G.R.E. Viewer, at the moment of identification of the static hand pose that matches the PSL sign “T”.

Algorithms	Avg. Recognition Rate (%)	
	User Lib.	Generic Lib.
Template Matching	53,6	38,3
Discrete Cosine Transform	50,6	42,5
Hu Moments	25,8	17,8
PGH	58,1	30,3
SSD	5,8	2,8
PGH-SSD Hybrid	21,9	13,3
CALI	-	32,5

Table 2. Testing Static Hand Poses for Static hand pose recognition algorithms for all the PSL symbols.

The hardware configuration used for the experiments, are described in Table 1, and an average of 6 frames per second were obtained during the trials. To aid the above described precision test process, an application was built (O.G.R.E. Viewer, Figure 7), which allows for the monitoring of the engine. The application enables:

- The visualization of the processed images, showing the detected hand contour, optionally with background subtraction.
- The selection of the pose recognition algorithm from a list of the 7 studied;

- The selection of the gesture and action XML configuration files;
- The identification of the recognised static pose and a tree to navigate in the context of triggered actions;
- The collection of static hand poses to add to generic hand pose library, use by the CALI algorithm.

The results presented in Table 2, show that Template Matching, Discrete Cosine Transform and PGH are, significantly, the best methods studied for robust shape recognition. Hu Moments are weak classifiers as well as Simple Shape Descriptors alone.

An attempt to improve PGH effectiveness was taken by creating a hybrid algorithm based in its integration with Simple Shape Descriptors. These can't robustly recognize individual symbols, but are useful when excluding some possibilities from the standard set of PSL. PGH would then be able to precisely pinpoint the performed pose from this shortened set. However, at the current stage, we were unable to effectively combine both algorithms own advantages, thus resulting so far on a poor hybrid method for shape recognition. The CALI technique, was able to obtain a success rate of 32,5% using a generic library (the technique supports a set of "generic" samples per pose). Although ranking below Template Matching and PGH, we believe that this technique has potential since it handles a training phase (which is inexistent in the other algorithms) and, therefore, in theory, could improve its performance by introducing in the library of the system, a larger number of better hand pose samples. However, the technique shows some limits, since not all the static pose shapes are adequate for recognition using this technique. Although the average recognition rate for the best algorithm (PGH) is less than 60%, when compared to similar state of the art systems having recognition rates up to 90%, we must consider that all possible symbols were included in the testing session. This means that highly correlated symbols have interfered in the process, thus causing false pose interpretations. When considered individually, we have identified a smaller set of Sign Language symbols which have a high standalone recognition rate (Table 2). When combined, and having selected an appropriate algorithm from the table above, the overall recognition rate can rise up to 90%. A carefully chosen subset (or a user defined set of creative poses), such as the ones depicted in Table 3, could be used in specific control functionalities. This was the approach taken in the application implemented in the "House of The Future", where the static hand poses shown in Figure 9, mostly related to Portuguese Sign Language Signs, were selected. After analysing the precision results, the Template Matching algorithm was the one used in the developed application we describe next, based on the algorithm good recognition precision results (Table 3).

5. APPLICATION OF GESTURE HCI FOR COMMAND AND CONTROL

By developing a close collaboration with the Portuguese Foundation of the Communications, namely with the

Algorithms	Symbols
Template Matching	1,3,5,7,B,D,H,K,L,N,P,T,V,W,Y
DCT	1,2,5,7,9,A,B,D,L,N,P,T,Y
CALI	1, 5, 7, L
Hu Moments	5, H, L
PGH-SSD Hybrid	1,C,Z
SSD	T

Table 3. Subsets of PSL symbols with average recognition rates higher than 80%, for the Template Matching, DCT, CALI, Hu Moments, PGH-SSD Hybrid and SSD algorithms.



Figure 8. Top: The application set-up as installed at the "House of the Future". Bottom: interacting via gestures with the house window characteristics.

group that supports the citizens with special needs, a demonstration application has been set in the "House of the Future" test-bed of the Communications Museum in Lisbon, Portugal (Figure 8), enabling gesture HCI for command and control. Several demonstrators were already in place, such as, accessible Internet browsing for persons with vision impairments or speech activated commands to control house appliances. After identifying the opportunity to deploy gesture activated commands, especially oriented to citizens with hearing disabilities or motor impairments, other than the movements of the hands, we have collected the overall requirements of our application, namely:

- The universe of static poses should be reduced to a minimum, easily memorised;
- Since the application will not have any link with the interpretation of PSL (Portuguese Sign Language), the correspondence between the identified static poses, when these are used in such Sign Language, and the application symbols, should be changed in order not to confuse the user that is acquainted with PSL.

We have selected six static hand poses (Figure 9), which show small correlation and high recognition rate (above 80%) in the precision tests (Table 1), that will be used to

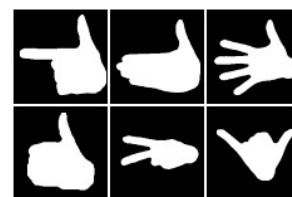


Figure 9. Six static hand poses of the left hand identified by our system, related to Portuguese Sign Language Signs. From top to bottom and left to right, we have the following signs: T, 1, Open Hand (not a sign), B, lying V and Y.

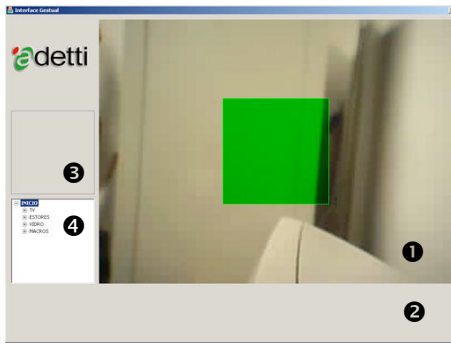


Figure 10. User's Interface. Field 1: Video Camera feedback; Field 2: Action Label; Field 3: Recognition feedback; Field 4: Actions Hierarchy Tree.

trigger user generated actions. The application commands (issued by gestures) were organised in a hierarchy manner, such that each hand pose would activate a menu option and, inside this menu, hand poses of the previous hierarchy level could be re-used.

Our O.G.R.E system architecture was integrated in a Home Network platform that controls home appliances, linked via ETHERNET, such as Set-Top-Box, TV Set, DVD player, Home lighting or Window Blinds developed by Portugal Telecom [Mar04]. A software module was developed to access a set of available Web Services that exposed the control of appliances of the house, for each interpreted and recognised gesture. By invoking these Web Services, it was possible to interface with the house equipment system command and control, therefore enforcing gesture commands for a set of selected equipments. A usage scenario in a living room, where a user interacts with home appliances (by means of selected static poses taken from the Portuguese Sign Language), was placed in operation. The application was installed in the child's room, to exemplify the control of the surrounding home environment (Figure 10). The user's interface is very simple in order to promote straightforward demonstrations, during museum tours inside the "House of the Future" (Figures 8 and 10). It consists in a full-screen window where 80% of its area shows feedback from the users hand gesture action in front of the camera. There is an initial calibration process when the application is launched in order to perform background subtraction. After system calibration the user can perform hand poses to activate actions that allow the



Figure 11. The application interface installed in the "House of the Future". The recognized static pose is presented to the user on the left, and the triggered action is indicated below (opening the window blinds).

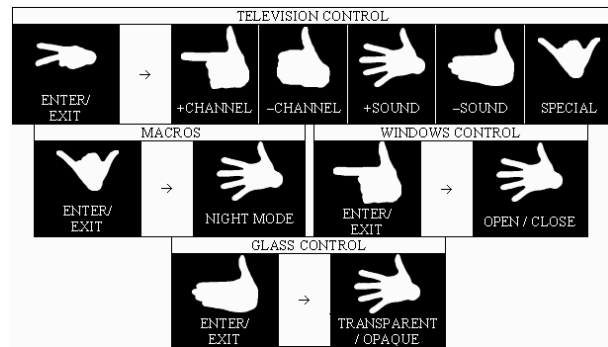


Figure 12. Available menus that can be browsed by performing specific hand poses.

command and control of house-hold devices. Figure 12 shows all available menu options. It is possible to control the TV set (select a channel up or down, turn the volume up or down or access pre-programmed channels), open or close the windows blinds, change the opacity of a polarised glass window and activate macros (set of simultaneous control actions inside the house, such as turn the light of or on or shutdown the windows blinds and night fall).When a drastic change of lightning or camera position occurs, the system launches an automatic recalibration process immediately (see section 4, Background Analysis).

6. USABILITY EVALUATION

The application was subjected to usability testing by 10 unpaid undergraduate students of both sexes, of the degrees of Engineering and Informatics at ISCTE in Lisbon, and five users of both sexes, which were support staff working in the "House of the Future. All the subjects had no apparent impairments. For this study, we have followed the model described in [Dias03]. The major aim of this testing phase was to understand how well adapted the system was to the needs and skills of the users. Three questionnaires were elaborated:

1. Questionnaire B.2, to assess the experience and opinion of subjects in relation to different user interface modalities, in relation to which our interaction technique could be compared.
2. Questionnaire B.3, filled after an audio-visual presentation, followed by a demo of the system in action, aimed at obtaining a first impression regarding the system functionality.
3. Questionnaire B.4, filled after the user trial, which followed a pre-established B.1 trial script.

The B.1 trial script guided the subject in a set of tasks that exercised his/her skills in executing all the gestures supported by the system. The subject was initially asked to browse the system menus, through static hand poses, that simulated the activation (in the lab), or activated (in the House), the remote control of the house devices, via gestures, such as the TV set (by increasing or decreasing the volume or snapping up and down the TV channels), the window binds (by opening or closing), or the window glass transparency (turning it transparent or opaque). The

subject was asked also to perform Simple Paths and Staged Paths [Dias04]. The answers to the Questionnaires were based in a quantitative scale of 7 levels, ranging from -3 (negative or a deficit assessment) to 3 (positive or even excessive assessment). Zero (0) was considered to be the neutral indifferent answer. The subjects were trained in the system concepts and in the tasks to be performed prior to the trials, which took a considerable amount of time. As mentioned, ten unpaid subjects of both sexes (but males in majority) were selected from the graduate and undergraduate courses in ISCTE, with ages ranging from 20 to 25 years of age, of the Engineering and Informatics courses. Five more exhibition guides in the “House of the Future” museum (mostly females of ages between 20 and 25 years of age), were also subjected to the same usability testing method. The results of questionnaire B.2, have showed that subjects were open to use “state-of-the-art” user interaction technologies, to assist in executing generic tasks. However, subjects were much more acquainted and, therefore, more biased towards mouse and keyboard interaction, than with other HCI modalities, such as gesture, although still favourable in trialling this last modality. The results of questionnaire B.3, have showed a high interest for the system, due to its high degree of novelty and unfamiliarity. When compared with other HCI modalities, subjects believed that it could be an alternative to data gloves. Static hand poses were considered to be easier than simple paths and these ones were believed to be more straightforward than staged paths. The global system impression was positive. Questionnaire B.4 results, have assessed the final opinion of the tested subjects. This could be compared with questionnaire B.3, where we could identify a decrease in the perceived ease of use of the system (producing physical gestures was more difficult than initially expected) and a small increase in the familiarity of use. In relation to B.3 results, the system was considered to be a viable alternative to remote commands of home equipment in addition to data gloves. Static hand poses were considered to be the simplest and easiest gesture to be executed, whereas the simple paths were more difficult and the staged paths were the ones that created the bigger difficulties in use. In general terms, the subject’s reaction to the system was positive and they have considered that the system was easy to use, especially when good lightning conditions were made available.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have described the different architectural modules of a hand gesture recognition engine based on computer-vision. The system is configured with XML specifications that describe the type of gesture to be recognized in a given context with a number of possible static hand poses available. The system was evaluated, regarding its precision in recognizing with success certain hand poses. An experiment was set-up, where a subject was issuing static hand poses of Spelled Portuguese Sign Language, to assess the robustness of various algorithmic alternatives to handle the sub-problem of shape recognition, present

in the hand pose understanding process. Our results have shown that Pair-wise Geometrical Histogram contour-based method, is the most effective in relation to the average symbol recognition rate metric, reaching the figure of 58.1% for the case of the own user library of symbols, followed by the Template Matching image-based method. If the test is only made with highly non-correlated symbols, the metric can rise up to 90%. In this case we have identified 15 symbols of PSL, whose recognition rate is higher than 80%. This means that these symbols can be used with success in gesture-based HCI tasks, where the basic gesture to issue is a static pose. This achievement is one of the original contributions of this paper, in addition to others, such as the integration of generic background removal with the gesture recognition technique, still achieving near real-time static hand pose recognition rates, (6 frames per second on a Pentium IV with 3 GHz CPU). We have also shown that our system can be generalized to application in general Human-Computer Interface tasks which require the static hand gesture recognition modality at interactive rates. A usage scenario in a living room, where a user interacts with home appliances (television, windows, lights, etc) by means of selected static hand poses, was developed and deployed in the “House of the Future” test-bed of the Communications Museum in Lisbon, Portugal, and it is still in operation at the time of writing of this paper. The application was subjected to usability testing by 10 users in our laboratory at ISCTE, and 5 more users inside the “House of the Future. In general terms, the subject’s reaction to the system was positive and they perceived the system as being easy to use, especially when good lightning conditions were made available. Usability testing with people with special needs (hearing and motor impairments) would also be an enriching asset for this work and is planned in the mid-term. As a natural continuation of our work, we aim at bimanual gesture recognition, hand feature extraction for finger recognition and occlusion treatment and face motion detection, using vision based approaches, by possibly assessing other 2D contour detection techniques, such as Zernik moments [Prokop 02]. Approaches enabling more flexible hand pose recognition by compensating the position of a moving camera can also be addressed. By generating time-based vectors of characteristics related to hand, finger and face gestures, we plan to develop a statistical model based in Hidden Markov Models [Park 06], that can be trained with sufficient input gesture data of the mentioned type, in order to create robust and real-time gesture recognition engines, that can be applied to “hard” long-term problems, such as full Portuguese Sign Language interpretation.

8. ACKNOWLEDGEMENTS

The authors would like to thank Eng^o Gonalo Areia, Eng^o Joel de Almeida, Dr^a. Isabel Manteigas, Sr. Jos Raposo (Fundação Portuguesa das Comunicaões), Eng^o Pedro Santos, Eng^o Bruno Marques, Eng^a Clara Cidade, Dr. Tiago Alves (Portugal Telecom), Professor Joaquim Jorge, Eng^o Manuel da Fonseca (INESC-ID), Eng^o Pedro Santos (Microsoft), Eng^o Rafael Bastos (ADETTI), for

their valuable help in specifying and supporting our application.

9. REFERENCES

- [Aguilar03] Aguilar, M., Joshua R., Hasanbelliu, E., "Facilitating user interaction with complex systems via hand gesture recognition". ACMSE'03. Knowledge Systems Laboratory, Jacksonville State University, 2003.
- [Bradski98] Bradski, G., "Computer vision face tracking for use in a perceptual user interface". Intel Technology Journal. Microcomputer Research Lab, Santa Clara, CA, Intel Corporation, 1998.
- [Bradski02] Bradski, G., "Intel Open Source Computer Vision library overview", Intel Labs, Intel Corporation, 2002.
- [Chaudhury00] Santanu Chaudhury, Subhashis Banerjeeb, Aditya Ramamoorthya, Namrata Vaswani. "Recognition of dynamic hand gestures". Journal of The Pattern Recognition Society. Department of Electrical Engineering and Department of Computer Science Engineering, IIT Delhi, 2000.
- [Davis99] Davis, L. S., Horprasert T., Harwood D. "A statistical approach for real-time robust background subtraction and shadow detection". Technical report, Computer Vision Laboratory University of Maryland, 1999.
- [Dias03] Dias M, Jorge J, Carvalho J, Santos P, Luzio J, "Usability evaluation of tangible user interfaces for augmented reality", IEEE International Augmented Reality Toolkit Workshop: pp 54-61, 2003, Proceedings of ART03, 2nd IEEE International Augmented Reality Toolkit Workshop, Waseda Univ., Tokyo, Japan, Oct 07, 2003.
- [Dias04] Dias, J. M. S., Nande P, Barata N, Correia A., "OGRE - Open Gestures Recognition Engine", in Araujo AA, Comba JLD, Navazo I, Souza AA, Eds., XVII Brazilian Symposium on Computer Graphics and Image Processing/II Ibero-American Symposium on Computer Graphics, SIBGRAPI/SIACG Curitiba, Brazil, Proceedings IEEE COMPUTER SOC: pp 33-40,, 17th-20th October 2004.
- [Eckhardt00] Eckhardt, U., Latecki, L., Lakämper, R.. "Shape descriptors for non-rigid shapes with a single closed contour". IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2000. Dept. of Applied Mathematics University of Hamburg, Germany, 2000.
- [Fonseca00] Fonseca, M. J., Jorge, J. A. "Using Fuzzy Logic to Recognize Geometric Shapes Interactively", Proceedings of the 9th Int. Conference on Fuzzy Systems, (FUZZ-IEEE'00), pp 191-196, San Antonio, USA, 2000.
- [Geoffrey98] Geoffrey E., Hinton S., Sidney Fels. "Glove-talk 2 - a neural-network interface which maps gestures to parallel formant speech synthesizer controls". IEEE Transactions on Neural Networks, Vol. 9, January 1998.
- [Gianino84] P. D. Gianino J. L. Horner. "Phase-only matched filtering". Journal of Applied Optics, 1984.
- [Heuer84] Heuer, J. Kaup, A., "Polygonal shape descriptors - an efficient solution for image retrieval and object localization", 34th Asilomar Conference on Signals, Systems and Computers. Siemens Corporate Technology, Information and Communications, Munich, Germany, 2000.
- [Hub 98] Faria, I. H., Falé, I., Viana, M. C., Pereira, C., "A Língua Gestual Portuguesa Como Um Sistema Linguístico: análise de alguns verbos", Proceedings of XIV Encontro Nacional da Associação Portuguesa de Linguística, University of Aveiro, 1998.
- [Kadous95] Kadous W.. "Grasp: Recognition of Australian Sign Language using Instrumented Gloves". Technical report, The University of New South Wales, Schools of Electrical Engineering and Computer Science and Engineering, October 1995.
- [Lu02] Guojun Lu, Dengsheng Zhang. "A comparative study on shape retrieval using Fourier descriptors with different shape signatures". 2002. Fifth Asian Conference on Computer Vision (ACCV02). Gippsland School of Computing and Information Technology, Monash University, Australia, 2002.
- [Maydt02] Maydt, J., Lienhart, R.. "An extended set of Haar-like features for rapid object detection", IEEE ICIP'2002, Intel Labs, Intel Corporation, 2002.
- [Mar04] Bruno Marques. Web services da casa do futuro - Especificação Técnica. Technical report, Portugal Telecom - Sistemas de Informação, March 2004.
- [Niwa02] Yoshinori Niwa, Kazuhiko Yamamoto, Terrillon, J., Pilpré, A. "Robust face detection and Japanese Sign Language hand posture recognition for human-computer interaction in an "intelligent" room". VI'2002. Office of Regional Intensive Research Project (HOIP), Softopia Japan Foundation, Faculty of Engineering, Gifu University, 2002.
- [OGRE3D] www.ogre3d.org
- [Park 06] Park, A., Lee, S., "Gesture Spotting in Continuous Whole Body Action Sequences Using Discrete Hidden Markov Models", in Sylvie Gibet, Nicolas Courty, Jean-François Kamp, Eds, Gesture in Human-Computer Interaction and Simulation, LNAI, Vol 3381: pp 100-112, 6th International Gesture Workshop, GW 2005, Revised Selected Papers, 2006.
- [Prokop 02] Prokop, R. J., Reeves, A. P. "A survey of moment-based techniques for unoccluded object representation and recognition". CVGIP Graphical models and Image Processing, 54(5): pp.438-460, 1992.
- [Shirai02] Yoshiaki Shirai, Nobuhiko Tanibata, Nobutaka Shimada." Extraction of hand features for recognition of sign language words". VI'2002. Computer-Controlled Mechanical Systems, Graduate School of Engineering, Osaka University, 2002.
- [Wirth02] Michael A. Wirth. "Shape analysis and measurement". University of Guelph. CIS*6320 Image Processing Algorithms and Applications. Computing and Information Science Biocomputing Group, 2002.

Avaliação de Interfaces Multimodais para Livros Falados Digitais com foco Não Visual

Carlos Duarte Teresa Chambel Hugo Simões Luís Carriço
LaSIGE / Faculdade de Ciências da Universidade de Lisboa
{cad,tc,hsimoes,lmc}@di.fc.ul.pt

Eduardo Santos, Guilherme Francisco, Sérgio Neves, Ana Catarina Rua, João Robalo, Tiago Fernandes
Faculdade de Ciências da Universidade de Lisboa
{i32097,i32169,i32536,i31586,i32059,i29059}@alunos.di.fc.ul.pt

Sumário

Este artigo apresenta um estudo de avaliação de interfaces multimodais para Livros Falados Digitais, tendo em conta diversos tipos de utilização, inspirados quer por diferentes características dos utilizadores quer pelas situações de uso. São apresentados: uma análise de trabalhos relevantes na área e requisitos a ter em conta; o Rich Book Player, um dispositivo de reprodução de livros falados digitais, multimodal e adaptativo; e um estudo de usabilidade das suas funcionalidades e capacidades de interacção, com ênfase em aspectos não visuais.

Palavras-chave

Livros Falados Digitais, Interfaces Multimodais, Interfaces Não Visuais, Acessibilidade, Usabilidade, Avaliação

1. INTRODUÇÃO

Com o desenvolvimento da sociedade da informação, os pressupostos sobre o utilizador típico têm vindo a alterar-se. Este já não é, necessariamente, uma pessoa que utiliza a tecnologia essencialmente no ambiente de trabalho e sem necessidades especiais de interacção. A convergência das tecnologias da informação, telecomunicações e electrónica de consumo, introduziu alterações radicais no mercado, colocando uma maior ênfase na componente residencial e na mobilidade, e consequentemente no tipo e contexto de uso das aplicações interactivas. Em face desta mudança de paradigma, o público alvo é bem mais diverso e heterogéneo, quer em termos de conhecimentos e envolvente cultural, quer de capacidades e contextos de utilização, e os dispositivos de acesso mais diversificados em termos de plataformas tecnológicas. Neste contexto, tornam-se cada vez mais relevantes os aspectos de acessibilidade e usabilidade para todos, em qualquer lugar e em qualquer momento [Stephanidis2000].

Os livros falados, tradicionalmente gravados em meios analógicos, disponibilizam versões de livros impressos, tornando-os acessíveis a pessoas ou contextos de utilização diversos do tradicional. As pessoas com deficiências visuais têm sido algumas das que mais têm beneficiado com esta tecnologia, apesar das limitações de usabilidade que estes livros apresentam. Em versão digital, estes livros podem disponibilizar um leque mais rico de capacidades, colmatando problemas de usabilidade encontra-

dos nas versões analógicas, incluindo melhores formas de navegação e a possibilidade de armazenar vários livros em dispositivos de menores dimensões. Com a possibilidade de integrar diversos meios, como o áudio, o texto do livro e imagens, os livros falados digitais podem ser mais versáteis e facilitar a disponibilização de múltiplas modalidades de interacção, o que pode ser atractivo para leitores com dificuldades perceptuais permanentes ou temporárias, como é o caso dos cegos, ou pessoas com outras deficiências visuais, motoras, ou outro tipo de dificuldade em lidar com versões impressas dos livros [Moreno2000]. As capacidades multimédia e multimodais destes livros podem ser exploradas para enriquecimento da experiência de leitura destas pessoas, mas também de pessoas sem necessidades especiais, ou em contextos situacionais que imponham temporariamente restrições em termos cognitivos, perceptuais e de atenção [Gazzaniga1998].

O objectivo do presente estudo consistiu na avaliação de interfaces multimodais de livros falados digitais, tendo em conta diversos tipos de utilização, inspirados quer por diferentes características dos utilizadores quer pelas situações de uso. Para tal, foram analisados trabalhos relevantes na área e requisitos a ter em conta. Foi ainda feito um estudo de usabilidade do Rich Book Player [Duarte2006], um dispositivo de reprodução de livros falados digitais, multimodal e adaptativo, desenvolvido

no contexto do projecto de investigação RiCoBA¹ – Rich Content Books for All.

Este trabalho resultou de uma colaboração entre elementos da equipa do RiCoBA, sendo um deles docente da disciplina de Interfaces Pessoa Máquina do 2º. Ano da Licenciatura em Eng. Informática da Faculdade de Ciências da Universidade de Lisboa, e dois grupos de três estudantes dessa disciplina. Cada um destes dois grupos incluiu um estudante invisual. A sua experiência como utilizadores de tecnologias com foco na acessibilidade, em particular com interfaces não visuais, aliada à experiência dos restantes colegas em interfaces visuais, e num contexto de aprendizagem sobre a concepção, desenvolvimento e avaliação de interfaces, contribuiu para um cenário mais rico de trabalho. Este estudo surgiu por um lado como um desafio, por outro como uma oportunidade, permitindo enquadrar estes alunos num contexto de aprendizagem tradicionalmente mais focado para as dimensões visuais da interacção.

Depois desta introdução, a secção 2 introduz os conceitos e apresenta os requisitos e o estado da arte que definem o contexto deste trabalho. A secção 3 descreve a aplicação de reprodução de livros falados digitais Rich Book Player, evidenciando os princípios subjacentes e as principais funcionalidades. A secção 4 apresenta a configuração dos testes de avaliação efectuados, sendo os seus resultados apresentados e discutidos na secção 5. O artigo termina com a apresentação das principais conclusões e identificação de direcções para desenvolvimentos futuros.

2. CONTEXTO

Esta secção introduz os conceitos, requisitos e estado da arte nos aspectos que definem o contexto em que o trabalho se insere.

2.1 Multimodalidade e Acessibilidade

As interfaces multimodais permitem que o utilizador tire partido do seu modo ou modos mais naturais de comunicação, em cada tarefa ou situação, numa interacção com o computador. Estas interfaces podem ser usadas em dispositivos de grandes ou pequenas dimensões [Cohen2004]. Outra das suas vantagens é a desambiguação mútua, podendo a informação de uma das fontes ajudar a resolver ambiguidades na informação, reduzindo erros [Oviatt1999]. Por exemplo, num livro falado, a visualização de uma palavra escrita pode ajudar a desambiguar o sentido de palavras homófonas.

Para tirar partido das capacidades de processamento dos utilizadores (incluindo a atenção, a memória de trabalho e a tomada de decisões), os designers de interfaces multimodais deverão conceber os seus sistemas de forma a maximizar as habilidades cognitivas e físicas dos seus utilizadores. Algumas directivas que apontam nesse sentido incluem: 1) evitar a apresentação desnecessária de informação em duas modalidades diferentes, em situações em que o utilizador tem que prestar atenção simul-

tânea às duas fontes para compreender a informação, evitando sobrecarga cognitiva [Kalyuga1999]; 2) Maximizar as vantagens de cada modalidade, para reduzir sobrecarga da memória nalgumas tarefas ou situações [Wickens1992]. Exemplos desta combinação incluem: o uso combinado de apresentação visual combinado com interacção manual de informação espacial; e o uso combinado de apresentação áudio combinada com interacção por voz.

As abordagens tradicionais à acessibilidade, consideradas de natureza reactiva, têm visado a criação de configurações alternativas de hardware e software que permitam aos utilizadores com necessidades especiais aceder a aplicações originalmente desenvolvidas para utilizações tradicionais. As duas abordagens tecnológicas para a acessibilidade são: a adaptação ao nível do produto, onde cada aplicação é tratada em separado e criada uma versão de acessibilidade alternativa; e a adaptação ao nível do ambiente, onde a intervenção é feita de forma a disponibilizar software e hardware que permitam que este, e consequentemente as aplicações nele executadas, sejam acessíveis por meios alternativos.

As Interfaces Unificadas, Unified User Interfaces [Stephanidis2000], disponibilizam um conjunto de metodologias, técnicas e ferramentas que permitem a adopção de um paradigma mais proactivo para lidar com os requisitos das pessoas e situações em que há necessidades especiais. O conceito base consiste na criação de artefactos com interfaces únicas que encapsulam elementos alternativos de interacção mais adequados a cada pessoa ou situação. Um exemplo de uma abordagem reactiva na concepção de livros falados seria deixar a leitura do livro a cargo de um leitor de ecrã, como os que são usados pelas pessoas com dificuldades visuais no acesso à maioria das aplicações. De acordo com uma abordagem proactiva, a aplicação que suporta os livros falados disponibiliza versões lidas por locutores humanos, ou tem a capacidade de sintetização de voz, concebidos e integrados na própria aplicação. Esta abordagem tem-se mostrado viável mesmo em interfaces de uso geral e grande escala, como é o caso do web browser AVANTI [Stephanidis1997] que, de acordo com os autores, tem o potencial de apresentar melhores resultados que as interfaces dedicadas, desenvolvidas explicitamente para utilizadores com necessidades especiais, quer em termos de acessibilidade e qualidade da interacção, quer em termos de eficácia e eficiência económica a longo prazo.

2.2 Livros Falados Digitais

Os Livros Falados Digitais (LFD) surgem como uma evolução dos livros falados analógicos, disponibilizando meios de interacção inspirados na forma como interagimos com livros impressos. Os livros falados, tradicionalmente gravados em meios analógicos, como as cassetes áudio, disponibilizam versões de livros impressos falados ou lidos por pessoas. Quando comparados com os livros impressos, os livros falados apresentam algumas limitações: sendo gravados sequencialmente, são limitados no acesso aleatório; Não é possível interagir com os

¹ hcim.di.fc.ul.pt/ricoba/

livros, marcando páginas ou efectuando anotações; Não é possível aceder à forma escrita das palavras, uma queixa frequente dos utilizadores; Apenas disponibilizam uma versão dos livros; As formas de lidar com notas de rodapé, lidas quando referenciadas ou fora da sequência de leitura do livro, não agradam a todos nem em todas as situações de leitura. Os livros falados digitais permitem maior flexibilidade na leitura, criar marcas de página, evidenciar porções de texto e fazer anotações. O texto do livro pode ser apresentado de forma sincronizada com a sua leitura em áudio, permitindo a localização e visualização da forma escrita das palavras lidas. Também é mais fácil localizar, saltar ou ler as notas de rodapé quando for mais oportuno.

As pessoas com dificuldades visuais têm sido das que mais têm beneficiado com o uso de livros falados, para quem os livros impressos são tipicamente inacessíveis, com a excepção das versões em Braille, nem sempre disponíveis e limitadas em capacidades. No entanto, estes livros têm outros adeptos, e nos países onde são mais divulgados, é mais frequente encontrar pessoas que ouvem livros falados por exemplo no carro, enquanto conduzem, ou em casa, por vezes enquanto desempenham outras tarefas. Alguns dos problemas reportados quer em livros falados quer em livros Braille para material estruturado [Aldrich1988, Arons1997, Petrie1996] incluem: a necessidade de ter mais de uma casete áudio ou mais de um volume com o livro Braille, para cada livro impresso; é frequentemente difícil e moroso navegar e localizar informação, especialmente em várias cassetes ou volumes; é difícil fazer uma leitura rápida de livros baseados em cassetes áudio.

O consórcio Daisy [Daisy] é uma entidade responsável pela publicação de normas e especificações decorrentes da investigação no domínio dos livros falados digitais dos últimos anos. A National Information Standards Organization (NISO), em cooperação com o consórcio Daisy, publicou em 2002 a norma actual de LFD, a ANSI/NISO z39.86-2002 [NISO2002], também conhecida como DAISY 3.0. A norma foca a estrutura e conteúdo necessários à produção de livros falados, mas não inclui especificações para os dispositivos e aplicações de leitura destes livros. O Playback Device Features List [NISO1999], um documento auxiliar criado durante o desenvolvimento da norma, descreve as principais características que estes deverão ter, incluindo por exemplo: a não dependência de um dispositivo visual, tabelas de conteúdo navegáveis, acesso a níveis mais finos de detalhe, facilidade de aceder directamente a um destino, capacidade de adicionar informação como destaques e anotações, leitura de anotações, estabelecimento de marcações, avanço e recuo rápidos, disponibilização de fala humana e electrónica. Este documento não tem um carácter normativo e não apresenta soluções específicas de concretização. Consequentemente, os leitores de livros desenvolvidos de acordo com as normas de LFD adoptam diferentes soluções para a apresentação e interacção com os livros falados, muitas delas sofrendo de proble-

mas de usabilidade significativos, condicionando muitas vezes a acessibilidade das audiências a que se destinam.

2.3 Dispositivos de Reprodução de Livros Falados Digitais

Como sugerido pela especificação ANSI/NISO z39.86-2002, um dispositivo de reprodução, ou leitor, de um LFD é multimodal, apresentando o texto e o áudio narrado de forma sincronizada. São identificadas várias situações em que a interacção multimodal pode ser benéfica para o utilizador. Por exemplo, o reconhecimento de fala pode ser usado como modalidade de entrada, e a presença de marcações de página podem ser feitas de forma visual ou auditiva. No entanto, a maioria dos leitores de livros não fazem uso de múltiplas modalidades, nem vão muito para além da narração sincronizada com o texto.

O DAISY Playback Software é descrito como o primeiro dispositivo de reprodução de livros falados digitais num PC [Morley1998a], apresentando melhorias sobre o dispositivo dedicado PlexTalk Player [Plextor] também para livros DAISY. O leitor para PC foi concebido para utilizadores avançados que requerem acesso rápido e eficiente a livros estruturados. Foi concebido para PC, como aplicação autónoma que não faz uso de leitores de ecrã adicionais; é dirigido a utilizadores cegos, com deficiências de visão, ou visuais, sendo orientado para interacção por teclado e rato, com monitores capazes de apresentar letras de diversas dimensões; e permite pesquisas de texto nos títulos. Os comandos foram concebidos para serem consistentes e fáceis de usar, através do teclado normal, do teclado numérico, ou através do rato. A avaliação efectuada identificou falta de informação estrutural em interacções não visuais, devido à falta de indicações sonoras de informação que visualmente se distingue por tipo de letra diferente. A navegação entre secções também foi considerada confusa, em especial nas secções iniciais ou finais em navegação circular. No entanto, a maioria dos comandos foram considerados fáceis de usar e úteis. Para resolver os problemas encontrados, os autores sugerem a inclusão de sons não falados, e a adição de novos comandos.

Em [Duarte2005] é feita uma avaliação baseada em heurísticas de oito leitores de livros falados digitais, desenvolvidos de acordo com as normas DAISY e ANSI/NISO, incluindo os leitores AMIS 1.3, Book Wizard Reader, EaseReader, eClipseReader, gh Player 2.0, TAB Player, TPB Reader e Victor Reader Soft. Foram avaliadas as capacidades de navegação, as possibilidades de personalização oferecidas, a implementação de marcas de livros e de anotações, e a utilização de diferentes modalidades. Muitos dos leitores analisados não seguem várias das recomendações da norma ANSI/NISO. A maioria destas limitações não afecta fortemente interacções visuais, mas a interacção não visual é fortemente penalizada. De notar ainda que a maioria destes leitores não seguem as recomendações sobre a combinação criteriosa de modalidades referida anteriormente, resultando por vezes em experiências de leitura excessivamente

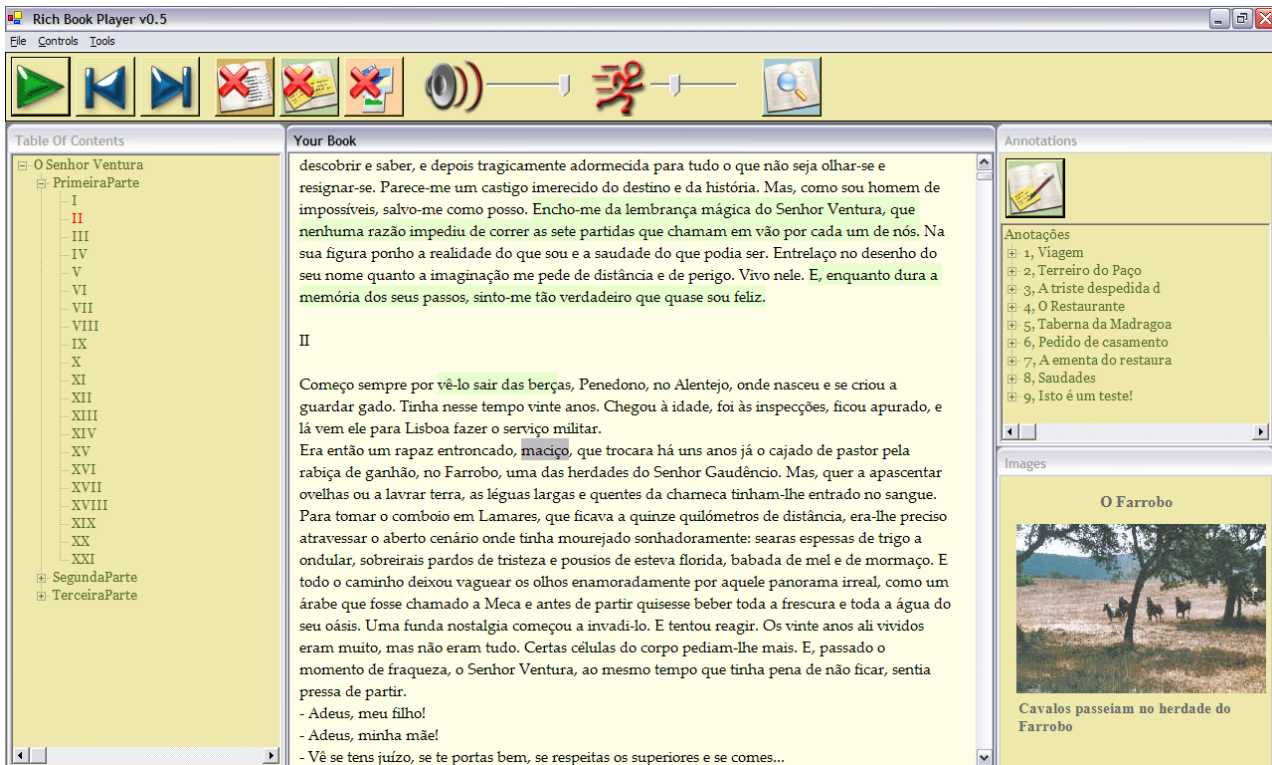


Figura 1 – O Rich Book Player, com todos os componentes visíveis.

complexas, com problemas de usabilidade e acessibilidade.

2.4 Outras interfaces não visuais

O estudo e desenvolvimento de interfaces não visuais tem sido efectuado noutras áreas relacionadas. Os sistemas hipermedia para invisuais [Petrie1996, Morley1998b] abordam aspectos semelhantes aos dos livros falados, com uma maior ênfase na definição, percepção e navegação de ligações, e operações de localização no hiperespaço. Um sistema que suporta interacção visual e não visual para a web é descrito em [Ntoa2005].

Um aspecto significativo na apresentação de livros é a apresentação de informação complexa de forma sumariada, como é o caso da informação contida numa tabela ou num mapa. Em [Kildal2006] é descrita uma interface não visual para tabelas, baseada na sonorização dos seus valores, por linha ou coluna, em diferentes níveis de detalhe. Em [Zhao2004] é apresentada a sonorização de informação geo-referenciada.

Em [Crispien1996] é apresentado um ambiente auditivo espacial que combina a apresentação espacial de informação com interfaces baseadas em gestos e fala. O utilizador é rodeado por um anel virtual contendo objectos de interacção auditivos, num diálogo de navegação multimédia. Os áudiojogos² constituem um exemplo de uma área de aplicação onde se exploram estes tipos de navegação, sendo o efeito de Doppler uns dos usados para percepção espacial.

² www.audiogames.net/

3. RICH BOOK PLAYER

Os livros falados combinam a apresentação visual e a narração de conteúdos literários de forma a melhorar quer a usabilidade, quer a acessibilidade a esses conteúdos. A combinação da representação áudio com o texto dos livros, que aparentemente seria apenas direccionada para utilizadores sem problemas de visão, permite oferecer funcionalidades, como a pesquisa, impraticáveis em livros com suporte apenas áudio, e que trazem benefícios acrescidos a utilizadores com baixa visão e cegos.

3.1 Concepção

Com o objectivo de tornar o alcance destes livros ainda mais abrangente, desenvolveu-se o conceito de Livro Falado Enriquecido [Carriço2003], que combina, para além do texto e do áudio, outros componentes que se encontrem disponíveis, e que podem incluir, por exemplo, imagens, vídeos, música, ou outros sons. Para além disso, os livros falados deverão oferecer aos seus utilizadores a possibilidade de criar marcações e anotar o conteúdo.

De modo a cumprir estes e outros requisitos enunciados anteriormente, desenvolveu-se uma aplicação, o Rich Book Player (Figura 1), que permite a reprodução de Livros Falados Digitais Enriquecidos. Esta aplicação foi desenvolvida com base na FAME (Framework for Adaptive Multimodal Environments) [Duarte2006]. A concepção da aplicação baseada nesta framework permitiu desenvolver, de raiz, uma aplicação que considera os diferentes aspectos de interacção e apresentação, aplicáveis à diversidade de potenciais utilizadores e situações de uso, indo assim ao encontro do conceito de interface

unificada [Stephanidis2000]. Desta forma, a aplicação será capaz de funcionar tanto em ambientes que disponham apenas de dispositivos que permitam entradas e saídas visuais, como em ambientes que disponham apenas de dispositivos áudio, como em ambientes onde seja possível combinar os dois tipos de dispositivos. É assim possível, com a mesma aplicação, oferecer as mesmas funcionalidades a uma diversidade de utilizadores que inclui os utilizadores sem deficiências, os utilizadores com baixa visão ou cegos, e os utilizadores com deficiências motoras mas que possam recorrer a comandos vocais.

Também ao nível dos conteúdos, a aplicação possibilita dois modos de operação, de modo a cobrir o máximo possível de situações de utilização. O modo de operação preferencial recorre a gravações áudio dos livros. O modo alternativo, a ser empregue em situações em que, por exemplo, não haja possibilidade de armazenar as gravações áudio, utiliza síntese de voz para transmitir o conteúdo.

Tanto para o reconhecimento, como para a síntese de voz empregou-se o Microsoft Speech 5.1. De notar que, quer para um, quer para outro, só se encontravam disponíveis motores de reconhecimento em Inglês, tendo sido por isso necessário adaptar a fonética dos comandos que eram fornecidos a ambos os motores, de modo a permitir o funcionamento da aplicação em Português. Por exemplo, para que o comando “início” pudesse ser reconhecido pelo motor de reconhecimento em Inglês teve de ser escrito como “ynyssiu”. Este factor poderá ter condicionado os resultados dos testes à frente apresentados, visto que se verificou uma evolução positiva na taxa de reconhecimento do reconhecedor com o avançar da utilização. De forma a minimizar este efeito, foi dada especial atenção à selecção dos comandos disponíveis para utilizar. A selecção dos comandos e do retorno a dar ao utilizador são aspectos fundamentais na concepção de interfaces baseadas em áudio.

3.2 Funcionalidades

Actualmente, o Rich Book Player disponibiliza as seguintes funcionalidades aos seus utilizadores:

Navegação – no Rich Book Player, a navegação pode ser realizada de diferentes formas:

- Através da selecção livre de um ponto de leitura no texto, sendo a narração continuada a partir desse ponto;
- Através da selecção do capítulo ou secção desejada na tabela de conteúdos;
- Através de comandos que permitem avançar (recuar) para a próxima (anterior) frase ou parágrafo ou secção ou capítulo;
- Através de comandos que permitem ir directamente para o início ou fim do livro;
- Através de comandos que permitem ir para a anotação seguinte ou anterior.

Marcações e anotações – o Rich Book Player permite aos seus utilizadores criarem marcações em qualquer parte do texto, bem como seleccionarem excertos do texto e associarem-lhes anotações. O utilizador pode posteriormente consultar as várias anotações criadas, bem como utilizá-las para navegar no texto. De cada vez que o utilizador fecha a aplicação, é criada, automaticamente, uma marcação que permite recomeçar a leitura desse ponto quando o utilizador regressar à aplicação. O conteúdo do livro que tenha sido anotado é apresentado com uma cor de fundo diferente para assinalar essa situação.

Apresentação de outros conteúdos multimédia – o Rich Book Player permite a apresentação de conteúdos multimédia associados a partes do livro. Esses conteúdos podem incluir imagens, músicas ou outros sons. O comportamento por omissão é avisar o utilizador sobre a presença desses conteúdos, deixando a este a possibilidade de os ver ou ignorar. Este comportamento da aplicação evolui de forma a reflectir as preferências do utilizador quanto à exibição destes conteúdos.

Pesquisa – é possível ao utilizador efectuar pesquisas no Rich Book Player. O utilizador pode navegar nos resultados da pesquisa, sendo possível avançar ou recuar pelos vários resultados retornados pela pesquisa. O texto pesquisado é apresentado com uma cor de fundo diferente, de forma a realçá-lo, bem como a frase em que está inserido, que é apresentada com outra cor de fundo, de forma a contribuir para uma melhor visualização e mais rápida identificação do resultado da pesquisa.

Controlo da narração – é permitido ao utilizador controlar a narração áudio do texto, através de comandos básicos como parar ou iniciar a leitura. É ainda possível controlar o volume da narração, bem como a sua velocidade.

Sincronização – o Rich Book Player apresenta um mecanismo de sincronização visual que permite acompanhar no ecrã a palavra que está a ser narrada. Essa palavra é destacada alterando a sua cor de fundo. O utilizador, se pretender, pode seleccionar como objecto de destaque a frase ou o parágrafo a que a palavra pertence. A unidade de sincronização também é adaptada pela aplicação aquando do aumento da velocidade de narração, visto que a velocidades mais elevadas, acompanhar visualmente a palavra que está destacada começa a exigir maiores esforços cognitivos aos utilizadores.

Caminhos alternativos de leitura – é possível definir caminhos alternativos à narração principal. Actualmente, a única possibilidade é definir como texto a ser lido o conteúdo que foi alvo de anotações. No futuro, outros mecanismos de suporte a esta funcionalidade serão incluídos, possibilitando ao utilizador marcar texto com diferentes categorias (análogo ao procedimento de sublinhar texto com diferentes cores) e definir caminhos de leituras a partir dessas marcações.

Disposição automática dos componentes visuais – o Rich Book Player apresenta visualmente quatro componentes: o conteúdo principal do livro, a tabela de conteúdos, as anotações, e um espaço para apresentação de outros con-

teúdos como imagens ou vídeos. Estes componentes podem surgir ou desaparecer, quer por ordem do utilizador, quer por decisão da aplicação. O utilizador pode alterar a distribuição espacial dos componentes, e o Rich Book Player automaticamente adapta a posição dos outros componentes, movendo-os e alterando o seu tamanho, de acordo com dois requisitos: ocupar o espaço que tem disponível, e minimizar os movimentos do componente que apresenta o conteúdo principal do livro. Este último requisito tem como objectivo limitar ao máximo a possível distração causada por movimentos deste componente, que será o seguido com mais atenção pelo leitor, visto que alterações do tamanho ou da posição do componente levam a uma alteração da distribuição do texto no ecrã, o que pode levar a uma desorientação temporária do leitor que se vê obrigado a encontrar novamente o texto que estava a ler.

Mecanismos de percepção – o Rich Book Player consegue transmitir aos seus utilizadores a posição em que se encontram dentro do livro de várias formas. Visualmente, essa informação é transmitida: 1) mudando na tabela de conteúdos, a cor correspondente à entrada do capítulo que está a ser lido; e 2) através da posição da barra de deslocamento vertical na janela do conteúdo principal do livro, de forma a ter sempre visível o texto a ser lido. Através de áudio, de cada vez que se entra dentro de um novo capítulo, a sua identificação é anunciada por síntese de voz. Para além disso, outra informação de percepção é transmitida pelo Rich Book Player, relativamente à presença de texto anotado ou de texto com outros conteúdos multimédia anexados. O utilizador é informado através de avisos visuais sob a forma de ícones a piscar, e de uma voz sintetizada que indica o tipo de conteúdo ou a existência de uma anotação. O texto que tenha sido anotado também é apresentado com uma cor de fundo diferente.

Modos de interacção – o Rich Book Player permite aos seus utilizadores optarem por utilizar o teclado e o rato, ou comandos de voz, como modalidades de entrada. As modalidades de saída incluem voz, gravada e sintetizada, texto, imagens, e sons.

4. CONFIGURAÇÃO DOS TESTES DE AVALIAÇÃO

A avaliação preliminar do Rich Book Player decorreu no laboratório de usabilidade do LaSIGE, Laboratório de Sistemas Informáticos de Grande Escala da Faculdade de Ciências da Universidade de Lisboa. Durante a realização dos testes, procedeu-se à captura dos comandos de voz utilizados no mesmo computador onde foi executada a aplicação. Outro computador capturou as saídas áudio e vídeo que eram apresentadas ao utilizador. Com uma câmara de vídeo capturou-se a actividade do utilizador durante o teste.

Durante os testes, foi utilizado o livro "O Senhor Ventura" da autoria de Miguel Torga, gravado previamente por uma locutora profissional.

Os testes foram efectuados em duas fases. Os resultados da primeira fase permitiram identificar aspectos a melhorar e conceber novos comandos para colmatar insuficiências detectadas, em especial na interacção não visual. Alguns foram introduzidos na aplicação antes da segunda fase de testes, outros foram testados utilizando a técnica do feiticeiro de Oz.

No início de cada sessão de testes, era apresentado ao utilizador o contexto de uso da aplicação, sendo-lhe depois permitido um período de familiarização com esta de cerca de dez minutos. De seguida, eram propostas ao utilizador um conjunto de tarefas que implicavam exercitar diversas funcionalidades da aplicação, desde as capacidades de navegação e pesquisa, até à criação de anotações. Depois de terminadas as tarefas, era pedido ao utilizador para preencher um questionário que cobria vários aspectos, como a utilidade das funcionalidades disponibilizadas, a satisfação com o desempenho da aplicação, e a facilidade de utilização.

Desta forma, foi possível durante uma sessão de testes conseguir três tipos de medidas através de três métodos de avaliação empírica [Jordan1998]. Os questionários permitiram obter medidas com origem nos utilizadores, a experiência controlada permitiu realizar observações, e os registos de utilização permitiram obter medidas de uso através das diferentes capturas empregues.

De referir ainda que, numa segunda fase de experiências, se empregou a técnica do feiticeiro de Oz para avaliar a introdução de novas funcionalidades, mais direccionadas para a interacção não visual, como a possibilidade de controlar o nível de retorno áudio oferecido pela aplicação, e a introdução de um comando de voz para pedir à aplicação para dizer em que ponto do livro (capítulo, secção e parágrafo) a leitura se encontra.

No total, realizaram-se sessões de testes com vinte utilizadores, alunos da Faculdade de Ciências da Universidade de Lisboa, sendo doze dos utilizadores do sexo masculino e oito do sexo feminino. Nenhum dos participantes no teste tinha deficiências visuais. Para poder avaliar as diferentes funcionalidades e possibilidades de utilização da aplicação, os testes foram executados em três cenários de uso distintos: um cenário em que o utilizador dispunha da possibilidade de utilizar todos os modos de entrada e saída visuais e áudio; outro em que o utilizador tinha disponíveis todos os modos de saída, mas permitindo apenas entradas por comandos de voz; e, por fim, um cenário em que a interacção visual não era utilizada quer para entradas quer para saídas. Desta forma foi possível avaliar o desempenho da aplicação nos diferentes cenários, e também observar quais as modalidades preferidas dos utilizadores quando dispõe de alternativas.

5. RESULTADOS PRELIMINARES

Os testes decorreram em duas fases. Uma primeira fase permitiu aos estudantes que organizavam os testes de avaliação familiarizarem-se com os procedimentos utilizados em experiências controladas, e realizar uma análise inicial aos resultados, de modo a fundamentar sugestões

de melhorias e alterações à aplicação. Uma segunda fase, realizada recentemente, permitiu aos mesmos estudantes avaliar as suas propostas, tendo sido empregue a técnica do feiticeiro de Oz, nos casos em que não foi possível introduzir em tempo útil na aplicação as alterações propostas. As duas fases foram separadas por um intervalo de duas semanas.

5.1 Primeira fase de testes

A primeira fase de testes incidiu principalmente na avaliação da facilidade de utilização e utilidade de algumas das funcionalidades oferecidas pelo Rich Book Player: navegação a partir da tabela de conteúdos, procura de palavras, criação de anotações, e controlo sobre a velocidade de leitura.

A navegação usando a tabela de conteúdos foi considerada normal ou fácil por 55% dos utilizadores que reportaram tê-la utilizado, recorrendo ao uso do rato. 18% consideraram a navegação na tabela de conteúdos difícil, e 27% não utilizaram a tabela de conteúdos para navegar. Quando são utilizados comandos de voz, os valores são semelhantes, registando-se, no entanto, um decréscimo na percentagem de utilizadores que consideram a navegação normal ou fácil, 46%, e um acréscimo nos que consideram a navegação difícil, 27%. A mesma percentagem de utilizadores reporta não ter utilizado comandos de voz para navegar na tabela de conteúdos. 73% dos utilizadores consideram muito útil ou indispensável esta funcionalidade, enquanto que os restantes 27% consideram-na pouco útil ou inútil.

A procura de palavras foi considerada de fácil utilização por 100% dos utilizadores que empregaram o rato e o teclado para a executar. Sendo que a procura baseada exclusivamente em comandos de voz não se encontra desenvolvida, devido à impossibilidade de dispor de um reconhecedor de voz com gramática livre, não é possível estabelecer uma comparação entre as duas modalidades para esta funcionalidade.

A procura de palavras foi considerada muito útil ou indispensável pela totalidade dos utilizadores.

A possibilidade de criação de anotações foi também avaliada. Quando empregando o rato e o teclado, 46% dos utilizadores consideraram-na uma tarefa de fácil execução, 27% de difícil execução e os restantes 27% não a utilizaram. Quanto recorrerem a comandos de voz, apenas 36,5% consideraram a tarefa normal ou fácil, a mesma percentagem de utilizadores que consideraram a tarefa difícil, enquanto que 27% não utilizou comandos de voz para efectuar esta tarefa.

A criação de anotações foi considerada muito útil por 64% dos utilizadores, e pouco útil pelos restantes 36%.

No que diz respeito à possibilidade de controlar a velocidade de narração, esta foi considerada de fácil utilização por 100% dos entrevistados, quando se recorre ao rato, e por 75% dos entrevistados quando se empregam comandos de voz. A totalidade dos entrevistados considerou esta funcionalidade como muito útil ou indispensável.

Ainda nesta fase, avaliou-se a utilidade dos mecanismos de percepção sonoros que alertam para a presença de anotações e imagens. Estes mecanismos consistiam numa voz sintetizada que emitia as palavras “anotação” ou “imagem” conforme a situação. 56% dos utilizadores consideraram os alertas para a presença das anotações muito úteis ou indispensáveis, tendo esse valor subido para 89% quando o alerta dizia respeito a imagens. Esta diferença pode ser explicada pelo facto de as anotações disporem de um mecanismo de percepção que as imagens não possuem, nomeadamente, a alteração da cor de fundo do texto que se encontra anotado. Dessa forma, os utilizadores normovisuais podem identificar prontamente qual o texto que está anotado, já se encontrando conscientes dessa situação antes da narração chegar a esse texto. Esta situação não se verifica para as imagens, sendo por isso o alerta mais valioso para o utilizador.

Foi ainda pedido aos utilizadores para classificarem o Rich Book Player numa escala de um a cinco, tendo sido a classificação média no final desta primeira fase de testes de 3,8.

5.2 Segunda fase de testes

Através de algumas entrevistas abertas com utilizadores que participaram nas sessões controladas, e através da análise das capturas de ecrã e das gravações vídeo dessas sessões, os estudantes participantes no estudo elaboraram um conjunto de funcionalidades que poderiam ser acrescentadas à aplicação, e prepararam novas sessões de testes, de forma a avaliar as suas propostas. Como as maiores limitações identificadas foram ao nível do retorno em situações de interacção não visual, as propostas também foram maioritariamente dirigidas nesse sentido.

Para a realização desta fase de testes, recorreu-se à técnica do feiticeiro de Oz, de forma a poder simular funcionalidades ainda não desenvolvidas.

Uma das propostas passou pelo aumento da quantidade de informação retornada pela aplicação após a execução de comandos de navegação. Em condições normais, o retorno indica o capítulo em que a leitura se encontra, de cada vez que há uma mudança de capítulo. Nas condições do teste, a informação passava a incluir indicações sobre o parágrafo, e a ser disponibilizada de cada vez que se navegava para um parágrafo diferente. Os resultados mostraram que a percentagem de utilizadores que consideram a situação de retorno normal agradável (86%) é superior à percentagem que considera a situação de retorno completo agradável (71%).

Um conjunto de outros comandos foi testado, de forma a perceber qual a sua utilidade. Um desses comandos permitia ao utilizador saber a sua localização no livro (capítulo e parágrafo) a qualquer altura. Segundo 86% dos entrevistados, este comando é muito útil, sendo pouco útil para apenas 14%. É de notar que este comando será particularmente útil em situações de interacção não visual, visto que nas situações em que há retorno visual, 83% dos utilizadores consideraram normal ou fácil perceber a sua localização no livro.

Outro comando que, segundo 57% dos utilizadores seria útil, é um comando para pedir à aplicação que sintetize uma lista das anotações criadas até ao momento.

Foram ainda testados comandos que permitem controlar outros aspectos da narração. Um comando de voz para retirar o som da narração foi considerado útil por apenas 29% dos utilizadores. Isto pode dever-se à existência de outros comandos que permitem controlar o volume, e ao facto de as pessoas preferirem ouvir a narração. Um comando para elevar a velocidade de narração para o valor máximo instantaneamente também só foi considerado útil por parte de 29% dos utilizadores. Isto pode justificar-se pelo facto de a narração na sua velocidade máxima ser de muito difícil compreensão, e pela existência de variados comandos de navegação, que dispensam dessa forma a necessidade de avançar na narração à velocidade mais rápida, que é necessária em meios que não suportam o acesso aleatório aos seus conteúdos.

De referir ainda que se manteve a tendência, já evidenciada durante as primeiras sessões de teste, de considerar a utilização da rato e teclado como mais fácil do que os comandos de voz.

No fim dos testes foi novamente pedido aos utilizadores que classificassem a aplicação na mesma escala de um a cinco, sendo que desta feita, com a introdução, ainda que simulada, de novas funcionalidades, a classificação média subiu para 3,9.

Numa perspectiva de análise aos próprios testes, será interessante notar a evolução que se verificou nos estudantes que organizaram e realizaram os testes. Esse factor foi mais notório durante a realização dos testes com a técnica do feiticeiro de Oz, em que a preparação para a realização dos testes e a coordenação entre os vários intervenientes evoluiu à medida que se foram desenvolvendo.

6. CONCLUSÕES

Este artigo apresenta os resultados de um estudo de avaliação de usabilidade de interfaces multimodais para Livros Falados Digitais. Os Livros Falados Digitais combinam a apresentação visual do seu conteúdo com a sua narração. A sua população alvo principal é composta por utilizadores com deficiências visuais, apesar de que, com desenvolvimentos que enriquecem os conteúdos e a sua apresentação, estes livros tenham diversificado os seus utilizadores. As características intrínsecas destes livros, tornam-nos particularmente aliciantes para exploração de uma plataforma de apresentação multimodal, que ofereça capacidades de interacção áudio e visuais, quer ao nível das entradas, quer ao nível das saídas. Assim, este artigo começou por apresentar um conjunto de recomendações para o desenvolvimento de dispositivos de reprodução de Livros Falados Digitais, baseadas no conceito de interface unificada. De seguida apresentou-se o protótipo de leitor utilizado nos testes de avaliação.

O leitor desenvolvido explora o uso de múltiplos meios e modalidades, disponibilizando funcionalidades e acessibilidade melhoradas. Através de capacidades adaptativas,

permite ainda dar resposta a diversos contextos de uso, ambientes de leitura e características dos utilizadores. Este leitor resolve grande parte das limitações encontradas nos leitores actualmente disponíveis, respeitando as recomendações apresentadas, e permitindo melhorar a experiência de leitura de utilizadores com diferentes tipos de capacidade e em diversos contexto de utilização.

Os resultados dos testes de avaliação preliminares realizados permitiram aferir a utilidade das funcionalidades disponibilizadas, bem como perceber dificuldades de utilização experimentadas. Neste aspecto, as maiores dificuldades prenderam-se com problemas com o desempenho do reconhecedor de voz utilizado, por este ser um motor de reconhecimento de Inglês, o que obrigou a uma adaptação da fonéticas das palavras a reconhecer. Apesar de tudo, com o treino resultante da utilização, o desempenho do reconhecedor foi melhorando ao longo dos testes. Foi ainda possível realizar uma série de experiências recorrendo à técnica do feiticeiro de Oz, que permitiram avaliar funcionalidades ainda não desenvolvidas no protótipo, principalmente relacionadas com a interacção não visual, como qual o nível de retorno áudio sobre a posição de leitura no livro, tendo sido concluído que a introdução de uma funcionalidade que permita transmitir ao utilizador a sua localização no livro seria muito útil.

Futuramente, a aplicação será ainda melhorada com os resultados destes testes de avaliação preliminares, estando planeada a realização de mais testes, direccionados para a avaliação dos mecanismos de adaptação incluídos no Rich Book Player.

7. AGRADECIMENTOS

O trabalho apresentado neste artigo é parcialmente baseado em investigação financiada pela Fundação para a Ciência e Tecnologia através do projecto POSC/EIA/61042/2004.

8. REFERÊNCIAS

- [Aldrich1988] Aldrich, F.K. e Parkin, A. Tape Recorded Textbooks for the Blind: A Survey of Producers and Users. *The British Journal of Visual Impairment*, 1(1), 3-6, 1988.
- [Arons1997] Arons, B. SpeechSkimmer: A System for Interactively Skimming Recorded Speech. *ACM Transactions on Computer-Human Interaction*, Vol 4, No 1, 3-38, March 1997.
- [Carriço2003] Carriço, L., Guimarães, N., Duarte, C., Chambel, T., and Simões, H. Spoken Books: Multimodal interaction and information repurposing. In Proceedings of HCI'2003, International Conference on Human-Computer Interaction, Crete, Greece, June 2003, 680-684.
- [Cohen2004] Cohen, P., e McGee, D. Tangible multimodal interfaces for safety-critical applications. *Communications of the ACM*, 47(1), 41-46, 2004.
- [Crispien1996] Crispien, K., e Fellbaum, K. A 3D-Auditory Environment for Hierarchical Navigation in Non-Visual Interaction. In Proceedings of

- ICAD'96, International Conference on Auditory Display, Palo Alto, California, November 4-6, 1996.
- [Daisy] Daisy Consortium. Welcome to the DAISY Consortium Web site. Retrieved June 12, 2006, from <http://www.daisy.org>
- [Duarte2006] Duarte, C., e Carriço, L. A Conceptual Framework for Developing Adaptive Multimodal Applications. In Proceedings of the 11th ACM International Conference on Intelligent User Interfaces (IUI 2006), pp. 132-139, Sydney, Australia, 2006
- [Duarte2005] Duarte, C., e Carriço, L. Users and Usage Driven Adaptation of Digital Talking Books. In Proceedings of the 11th International Conference on Human-Computer Interaction (HCI 2005), Las Vegas, Nevada, USA, 2005.
- [Gazzaniga1998] Gazzaniga, M. S., Ivry, R. B., e Mangun, G. R. Cognitive Neuroscience – the Biology of the Mind. W. W. Norton & Company, 1998.
- [Jordan1998] Jordan, P. W. Methods for Usability Evaluation. In An Introduction to Usability (pp. 51-80). London - Bristol: Taylor & Francis.
- [Kalyuga1999] Kalyuga, S, Chandler, P, e Sweller, J. Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology*, 13, 351-371, 1999.
- [Kildal2006] Kildal, J., e Brewster, S.A. Non-Visual Overviews of Complex Data Sets. In Proceedings of ACM CHI' 2006, Montréal, Québec, Canada, April 22-27, 2006.
- [Moreno2000] Moreno, R., e Mayer, R. E. A coherence effect in multimedia learning: The case for minimizing irrelevant sounds in the design of multimedia instructional messages. *Journal of Educational Psychology*, 97, 117-125, 2000.
- [Morley1998a] Morley, S. Digital Talking Books on a PC: A Usability Evaluation of the Prototype DAISY Playback Software. In Proceedings of ASSETS'98, the Third Annual ACM Conference on Assistive Technologies, Los Angeles, CA, 1998.
- [Morley1998b] Morley, S., Petrie, H., O'Neill A., and McNally, P. Auditory Navigation in Hyperspace: Design and Evaluation of a Non-Visual Hypermedia System for Blind Users. In Proceedings of ASSETS'98, the Third Annual ACM Conference on Assistive Technologies, Los Angeles, CA, 1998.
- [NISO2002] NISO. Specifications for the Digital Talking Book. Retrieved June 12, 2006, from <http://www.niso.org/standards/resources/Z39-86-2002.html>
- [NISO1999] NISO. Digital Talking Book Player Features List. Retrieved June 12, 2006, from <http://www.loc.gov/nls/z3986/background/features.htm>
- [Ntoa2005] Ntoa, S., e Stephanidis, C. ARGO: A System for Accessible Navigation in the World Wide Web. *ERCIM News*, 61, April 2005.
- [Oviatt1999] Oviatt, S. Mutual disambiguation of recognition errors in a multimodal architecture. In Proceedings of Conference on Human Factors in Computing Systems, 576-583, Pittsburgh, Pennsylvania, USA, May 1999.
- [Petrie1996] Petrie, H., Morley, S., McNally, P. O'Neill, A-M. e Majoe, D. Initial Design and Evaluation of an Interface to Hypermedia Systems for Blind Users. In Proceedings of Eighth ACM Conference on Hypertext, (Hypertext'96), ACM, 48-56, 1996.
- [Plextor] Plextor Co. Ltd, Japan. <http://www.plextor.com/about.htm#JAPAN>
- [Stephanidis2000] Stephanidis, C. Universal Access Through Unified User Interfaces. In Proceedings of 15th Technology and Persons with Disabilities Conference, Los Angeles, CA, USA, March 20-25, 2000.
- [Stephanidis1997] Stephanidis, C., Paramythis, A., Karagiannidis, C., and Savidis, A. Supporting Interface Adaptation: the AVANTI Web Browser. In Proceedings of the 3rd ERCIM Workshop on User Interfaces for All, 1997.
- [Wickens1992] Wickens, C. Engineering Psychology and Human Performance. Harper Collins, 1992.
- [Zhao2004] Zhao, H., Plaisant, C., Shneiderman, B., and Duraiswami, R. Sonification of Geo-Referenced Data for Auditory Information Seeking: Design Principle and Pilot Study. In Proc. International Conference on Auditory Display (ICAD), 2004.

Controlo Miográfico de Dispositivos Móveis para Tetraplégicos

Tiago João Vieira Guerreiro Joaquim Armando Pires Jorge
INESC-ID, Instituto Superior Técnico / Universidade Técnica de Lisboa
Rua Alves Redol, 9, 1000-029 Lisboa
{tjvg,jaj}@immi.inesc.pt

Abstract

Este artigo apresenta a Electromiografia como uma interface para controlo de dispositivos móveis por tetraplégicos. Apesar de existirem actualmente várias tecnologias que possibilitam o controlo de computadores pessoais por parte de indivíduos com esta patologia, o mesmo não se reflecte num contexto móvel. Observando este panorama, é urgente aumentar as capacidades desta população neste contexto, possibilitando maior autonomia e consequente aumento comunicativo. Apresenta-se o trabalho realizado, que contempla um protótipo para controlo do ambiente de trabalho através de contracções musculares, e o trabalho futuro, cujas fases essenciais são a migração da aplicação para um dispositivo móvel e avaliação com utilizadores representativos da classe alvo do sistema.

Keywords

Acessibilidade, Tetraplegia, Electromiografia, Interação, Mobilidade, Comunicação.

1. INTRODUÇÃO

O estudo e desenvolvimento de interfaces não-convencionais para acesso a dispositivos electrónicos tem particular interesse para pessoas que apresentem incapacidade física de utilizar os modos tradicionais. Indivíduos com tetraplegia são um sub-conjunto da população incapaz de interagir com computadores através do teclado e de dispositivos apontadores comuns. É assim essencial estudar e desenvolver novas interfaces que permitam oferecer a esta classe de utilizadores a capacidade de realizar tarefas no computador a um nível semelhante ao possibilitado a um utilizador totalmente capaz. Actualmente, existem soluções bastante satisfatórias que possibilitam o uso de computadores pessoais por parte de doentes com patologia neuro-muscular grave, como é o caso da tetraplegia.

1.1 Motivação

A acessibilidade universal a dispositivos electrónicos é um tema de investigação bastante activo, onde continua a haver muito trabalho a desenvolver. Em particular, no caso dos doentes com patologia neuro-muscular grave, existem alguns sistemas que permitem aumentar o controlo de computadores, como são o caso dos *Trackers*, *BCIs*, *Cyberlink*, entre outros. O seguimento de características na face dos utilizadores (*Tracking*) é a tecnologia com maior sucesso para controlo do computador por utilizadores tetraplégicos, conseguindo mapear o movimento facial ou ocular em movimentos do dispositivo apontador. O “Eye-Tracking” baseia-se no seguimento, através de um sistema de captura especializado para o efeito, dos movimentos dos olhos, mais propriamente da íris. O seguimento facial, com ou sem marcas, consiste no controlo do ponteiro do rato

através de movimentos da face [Figueiredo05]. A segunda abordagem apresenta vantagens sobre o seguimento ocular pois dispensa um sistema evoluído de captura (bastante caro), sendo também mais robusta no que respeita a movimentos involuntários. Por outro lado, o seguimento facial necessita de um controlo efectivo de movimentos da face, o que nem sempre é possível. Outro exemplo de interacção por movimento ocular é usando electrooculografia, que consiste na recolha, através de eléctrodos, do sinal eléctrico gerado pelo movimento dos olhos. É possível medir movimentos nos dois eixos podendo assim controlar uma interface de forma relativa, ao contrário dos sistemas de seguimento, onde o mapeamento é absoluto [Chen03]. As interfaces cérebro-máquina apresentam-se como mais interessantes em casos onde a função muscular é nula, como acontece com o síndrome de Locked-in [Lehtonen02]. Nestes casos, o único canal de comunicação possível é o cérebro, que se encontra intacto. Nos últimos anos têm sido realizados avanços prometedores nesta área mas é ainda difícil oferecer o controlo de dispositivos com alguma complexidade de forma autónoma, utilizando apenas o sinal electroencefalográfico [Pfurtscheller06]. O sistema Cyberlink é um exemplo de sucesso do que se pode atingir ao juntar modalidades, neste caso, EOG, EMG e EEG [Marler04]. A electromiografia é utilizada também em sistemas de controlo isoladamente, sendo possível adaptar o sistema e as capacidades do mesmo à capacidade residual que o utilizador detenha [Barreto99] [Felzer05] [Moon04]. Apesar do controlo de computadores fixos poder ser melhorado, já é possível exercer um controlo satisfatório dos mesmos, sendo até possível aumentar o controlo de outros dispositivos

através deste. Por outro lado, torna-se uma prioridade oferecer o controlo de dispositivos móveis a utilizadores que não o conseguem fazer autonomamente, visto actualmente não existirem soluções neste contexto. A nível móvel, os resultados são escassos devido aos problemas técnicos que as modalidades apresentam, quer seja pela instabilidade do movimento e tamanho dos dispositivos, no caso do seguimento, quer seja pela intolerância ao ruído (EEG). O trabalho apresentado procura ultrapassar esta falha, possibilitando a utilizadores com dificuldades motoras graves (ex: tetraplégicos) o controlo dos seus dispositivos móveis. Temos como objectivo permitir a utilizadores tetraplégicos o controlo das principais funções do telemóvel como a realização/atendimento de chamadas, leitura/envio de mensagens escritas e gestão de contactos pessoais.

1.2 Abordagem proposta

A nossa proposta para o controlo de dispositivos móveis baseia-se no processamento de sinais miográficos. Através de eléctrodos de superfície é possível captar a activação dos músculos monitorizados, podendo assim associar eventos a determinadas contracções ou movimentos. Com esta solução, é possível determinar a existência de contracções voluntárias possibilitando assim a utilização do sistema sem eventos inesperados. Esta solução é também independente do ambiente envolvente sendo imune a ruído, movimento ou interferências causadas por outros indivíduos (ao contrário do Reconhecimento de Fala, EEG). Comparativamente a outros sinais fisiológicos (ex: EEG, EOG), o sinal electromiográfico apresenta a melhor relação sinal/ruído e maiores amplitudes [Moon04], o que também facilita o seu processamento, tornando-se um bom candidato para o controlo voluntário de dispositivos. O número de músculos voluntariamente contraídos é amplo possibilitando assim diferentes pontos de recolha, inclusive em casos de incapacidade de movimento da cabeça. A colocação dos eléctrodos é então efectuada consoante a lesão apresentada sendo o pescoço, maxilares ou as zonas temporais (perto dos olhos), observadas como boas opções.

2. PROTÓTIPO

A primeira fase do trabalho apresentado neste artigo foca-se no estudo da electromiografia e consequente análise de sinal miográfico. O protótipo desenvolvido teve como objectivo o estudo e validação da electromiografia para controlo de dispositivos. Assim, apesar do sistema ser dirigido para dispositivos móveis, esta versão foi desenvolvida para computadores fixos. Este protótipo que detém a capacidade de processamento requerida, detectando activações musculares em qualquer ponto voluntariamente contraído. Para utilização, demonstração e avaliação do protótipo, foi criado um mecanismo de emulação de eventos do sistema operativo. Assim, o utilizador pode ter associada uma contracção a qualquer evento do teclado ou do rato. Este sistema permite o controlo do ambiente de trabalho, podendo ser

lançada e controlada qualquer aplicação. Nesta secção, descrevemos em maior pormenor o sinal fisiológico utilizado bem como a arquitectura e utilizações do sistema.

2.1 Electromiografia

Electromiografia (EMG) define-se como o estudo da função muscular através da análise dos sinais eléctricos gerados durante contracções musculares. As contracções musculares são antecedidas por correntes electroquímicas que percorrem as membranas das fibras, gerando uma diferença de potencial entre as zonas activas e as zonas inactivas. Esta diferença de potencial consegue, devido às propriedades condutoras dos meios biológicos, ser captada na superfície do corpo humano através de eléctrodos [DeLuca97].

2.1.1 Aparelho de Electromiografia [Gamboa04]

O aparelho de electromiografia utilizado recolhe amostras a uma frequência de 1000Hz em 5 canais independentes. É relativamente pequeno (14cm*8cm*4cm) e pode ser facilmente adicionado a uma cadeira eléctrica (Fig. 1). O aparelho é portátil e comunica com o dispositivo de processamento através de uma interface bluetooth.



Figura 1 - Aparelho de Electromiografia

2.1.2 Interação Miográfica

A utilização recorrente da electromiografia em áreas relacionadas com a Medicina levou a um grande investimento científico para melhorar os procedimentos de análise e processamento do sinal miográfico. Este avanço culmina com a possibilidade de utilizar aparelhos electromiográficos portáteis que comunicam sem fios com um sistema de processamento. Esta portabilidade possibilita a qualquer indivíduo o transporte e uso de um aparelho deste tipo com grande aceitação social [Constanza05]. O tamanho e a referida portabilidade levam ao uso da electromiografia na interacção pessoa-máquina, existindo trabalho nas áreas de Acessibilidade, Robótica, Computação Móvel, Reconhecimento de Gestos, entre outras. Nenhum dos trabalhos encontrados e referidos nesta secção aborda a acessibilidade a dispositivos móveis mas são referenciados pois baseiam-se na utilização do sinal miográfico para controlo de dispositivos. [Roy94] apresentam uma interface pessoa-máquina baseada em gestos para pessoas com graves limitações a nível motor e oral devido a paralisia cerebral. Este trabalho apresenta resultados motivadores diferenciando gestos aparentemente iguais, através do uso de redes neuronais. Vários outros trabalhos na área utilizam algoritmos de classificação e reconhecimento de padrões para diferenciar gestos [Crawford05], conseguindo diferenciar, por exemplo, mais de 6 movimentos

efectuados por uma mão. Os resultados obtidos são particularmente relevantes no controlo de próteses [Soares03]. No nosso trabalho, visto ser possível obter os resultados esperados com técnicas de processamento de sinal em tempo real, preferimos não utilizar algoritmos que necessitem de fases de treino nem calibrações adicionais. No entanto, no futuro, poderá ser interessante, caso exista motivação para isso, aumentar o leque de opções através de mecanismos de treino e classificação. [Barreto99] introduzem um sistema que procura oferecer a utilizadores com graves limitações motoras a possibilidade de apontar e seleccionar num computador pessoal. Este sistema associa movimentos faciais ao controlo do rato, sendo bastante semelhante ao "Tongue Point" [Salem97] mas utilizando sinal miográfico. Este trabalho e outros que lhe sucederam são a base da nossa investigação. Na área da mobilidade, [Constanza05] dá ênfase à utilização da electromiografia como uma técnica de interacção móvel. O trabalho introduz o EMG como uma interface íntima e subtil, que pode ser utilizada sem afectar o ambiente envolvente. O nosso projecto aceita esta ideia e aumenta-lhe complexidade, focando em questões de Acessibilidade e aumentando o controlo efectuado sobre os dispositivos, que na abordagem de [Constanza05] é bastante minimalista. O trabalho apresentado em [Guerreiro06] apresenta a electromiografia com uma interface *wearable* de uso diário. A recente, mas vasta, utilização da electromiografia na interacção pessoa-máquina, apresenta-a como uma interface robusta, versátil e que, no contexto deste trabalho, oferece a possibilidade de aumentar a qualidade de vida de indivíduos com necessidades especiais.

2.2 Processamento de Sinal

De forma a obter alguma informação sobre a activação muscular é necessário processar os sinais recolhidos. O processamento realizado é composto por duas partes essenciais, uma de pré-processamento e outra de suavização (Figura 2). Posteriormente, através do sinal suavizado é possível detectar, em tempo-real, momentos de activação muscular e realizar acções de acordo com a acção muscular realizada.

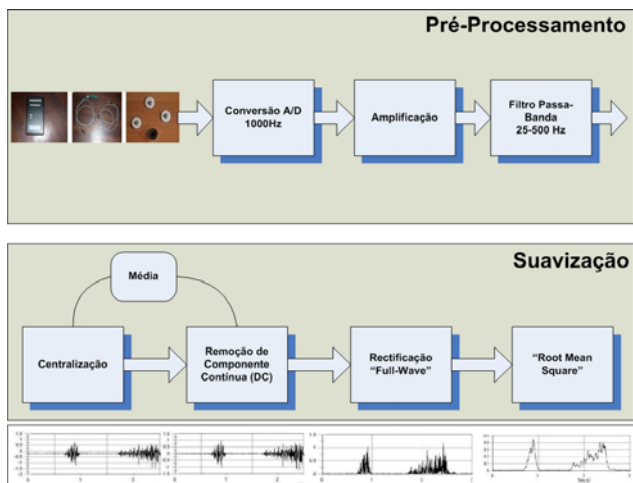


Figura 2 - Processamento de Sinal

Pré-Processamento

A fase de pré-processamento é realizada a nível de hardware, consistindo na amplificação do sinal, amostragem a uma frequência de 1000Hz e aplicação de um filtro passa-banda (25-500Hz) que remove frequências sem interesse a nível miográfico.

Suavização

Para conseguir detectar activações no sinal, é necessário suavizar o sinal recebido. O sinal recebido pelo aparelho de electromiografia tem uma gama de valores entre 0 e 4096, tendo este que ser ajustado, visto, na realidade, o sinal oscilar entre valores negativos e positivos. A centralização é uma operação muito básica e consiste em subtrair o valor de base (2048) ao sinal obtido. Em seguida, acrescenta-se o valor ao conjunto de valores já recebidos e com a média calculada sobre estes, retira-se a componente DC, que normalmente aparece no sinal EMG. Depois, executa-se a rectificação da amostra. A rectificação da curva é uma operação normalmente utilizada de forma a permitir a posterior integração do sinal, já que transforma uma curva com valores positivos e negativos, de média igual a zero, numa curva de valores absolutos, todos positivos. Existem duas formas de rectificar a curva: eliminando os valores negativos ("half-wave rectification") ou adicionando-os aos positivos ("full-wave rectification") [Correia92]. O último processo é preferível dado manter toda a energia do sinal e foi o adoptado. Finalmente, é necessário suavizar o sinal, tendo sido aplicado uma média quadrática (*Root Mean Square*) sobre a janela deslizante. O nosso sistema detecta ativações musculares através de uma abordagem onde o limiar é estimado como um múltiplo h do desvio-padrão [Staude01].

2.3 Cenário de Aplicação

O sistema de processamento desenvolvido permite detectar activações musculares possibilitando assim a execução de acções quando estas são detectadas. Assim, criámos uma aplicação que permite associar acções (do sistema operativo ou acções contextuais) a determinados padrões de acção muscular, oferecendo a possibilidade de controlo de qualquer aplicação desde que o número de eventos a emular seja limitado.

Controlo de Ambiente de Trabalho: Associando contracções musculares aos eventos do dispositivo apontador (4 direcções + 1 botão) é possível controlar, quase na totalidade o ambiente de trabalho. Este protótipo foi testado com utilizadores sem dificuldades motoras, com resultados que indicaram a possibilidade do controlo miográfico diário de interfaces [Guerreiro06]. Posteriormente, o sistema foi testado com utilizadores tetraplégicos (Figura 3), reduzindo o número de canais, mantendo-se as características apresentadas nos testes realizados a utilizadores com capacidade total. Este controlo pode ser realizado através de movimentos do pescoço, queixo ou um simples *piscar de olho*.

Dasher: Um dos objectivos do trabalho apresentado é a possibilidade de escrita de mensagens num telemóvel por



Figura 3 – Utilizador a interagir com o sistema

utilizadores tetraplégicos. A introdução de caracteres é a função que exige maior reflexão, sendo tradicionalmente utilizado um teclado no ecrã para o efeito, podendo ou não ser utilizados mecanismos de varrimento. Esta seria uma das opções mas procurámos obter maior desempenho e assim, idealizámos uma sinergia entre o protótipo apresentado e a aplicação Dasher [Ward00], que permite grande desempenho na escrita com movimento bidireccional ou unidireccional. O Dasher é uma aplicação de introdução de texto baseada numa técnica de zooming, com um sistema preditivo subjacente (Figura 4). Através desta sinergia, possibilitamos a escrita de texto com necessidade de apenas dois canais. Assim, o utilizador necessita, no limite, de apenas dois canais de entrada (cima e baixo, por exemplo) para navegar pelo “mar de letras” do sistema Dasher e cumprir a tarefa de escrita.

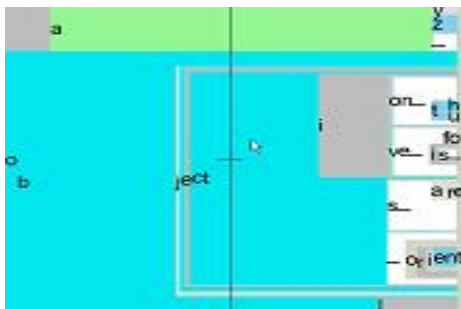


Figura 4 - Dasher

3. CONCLUSÕES

Apresentámos um trabalho em desenvolvimento que tem como objectivo oferecer a utilizadores tetraplégicos o controlo de dispositivos móveis. O protótipo actual permite a qualquer utilizador o controlo de aplicações num computador pessoal, validando o controlo de dispositivos através de contrações musculares. Apresentámos uma sinergia entre o Dasher e o nosso protótipo que permite a inserção de texto de forma eficiente, que adoptaremos. O próximo passo prende-se com a avaliação com utilizadores tetraplégicos. Esta avaliação será realizada tendo em conta o objectivo do trabalho, focando nas tarefas de utilização do telemóvel. Atingindo o controlo de um dispositivo móvel, tentaremos aumentar este controlo dando a hipótese de, através deste, controlar outros dispositivos.

4. REFERENCES

- [Barreto99] Barreto, A. et al.. A real-time assistive computer interface for users with motor disabilities. SIGCAPH Comput. Phys. Handicap.,(64):6–16, 1999.
- [Chen03] Chen, Y.. Design and evaluation of a human-computer interface based on electrooculography. Master’s thesis, Case Western Reserve Univ., 2003.
- [Correia92] Correia, P. et al.(1992). Estudo da funcao neuromuscular com recurso a electromiografia.. Technical report, Faculdade de Motricidade Humana.
- [Constanza05] Costanza, E. et al.. Toward subtle intimate interfaces for mobile devices using an emg controller. In CHI ’05, pages 481–489, NY, USA. ACM Press.
- [Crawford05] Crawford, B. et al. (2005). Real-time classification of electromyographic signals for robotic control. In AAAI, pages 523–528.
- [DeLuca97] De Luca, C.(1997). Surface electromyography: Detection and recording. Tech report, Delsys Inc.
- [Felzer05] Felzer, T. and Nordmann, R.. How to operate a pc without using the hands. In ASSETS 2005.
- [Figueiredo05] Figueiredo, L. and Raimundo, J.. Magic key. IADIS Ibero-Americana WWW/Internet, 2005.
- [Gamboa04] Gamboa, H. et al. (2004). Wireless physiologic data acquisition platform. Tech report, IT.
- [Guerreiro06] Guerreiro, T. and Jorge, J.. EMG as a Daily Wearable Interface. GRAPP, February 2006.
- [Marler04] Marler, D. M. Cyberlink: Computer access for persons identified with multiple disabilities. Technical report, California state university, northridge, 2004.
- [Moon04] Moon, I et al. A novel EMG based human-computer interf. for persons with disability. ICM’04
- [Lehtonen02] Lehtonen, J. EEG-based BCI. Master’s thesis, Helsinki Univ. of Technology, 2002
- [Pfurtscheller06] Pfurtscheller, G. et al.. Walking from thought. Brain Research, 1071:145–152, 2006.
- [Roy94] Roy, D. M. et al.. (1994). Gestural human-machine interaction for people with severe speech and motor impairment due to cerebral palsy. In CHI ’94
- [Staude01] Staude, G. et al.(2001). Onset detection in surface EMG signals: A systematic comparison of methods. Journal on App. Signal Process.,2:67–81.
- [Salem97] Salem, C. and Zhai, S. (1997). An isometric tongue pointing device. CHI ’97, pages 538–539, NY.
- [Soares03] Soares, A. et al. (2003). The development of a virtual myoelectric prosthesis controlled by an emg pattern recognition system based on neural networks.
- [Ward00] Ward, D. et al. (2000). Dasher - a data entry interface using continuous gestures and language models. In *UIST*, pages 129–137.

JavaSketchIt2 – Using Relational Adjacency Grammars for Visual Syntax Parsing

Filipe M Garcia Pereira
IMMI/INESC-ID/IST/UT
R. Alves Redol, 9, 1000Lisboa
fmgp@mega.ist.utl.pt

Manuel João Fonseca
IMMI/INESC-ID/IST/UTL
R. Alves Redol, 9, 1000 Lisboa
mjf@inesc-id.pt

Joaquim A. Jorge
IMMI/INESC-ID/IST/UTL
R. Alves Redol, 9, 1000 Lisboa
jorgej@acm.org

Abstract

This paper presents a project which expands on JavaSketchit [8], which parsed sketches of user interfaces, identifying gestures by using CALI [6], and then used topological relations and relational constraints [4] to recognize pairs of gestures as widgets in a Java graphical user interface. JavaSketchit was able to export the functional source code for the prototype in Java. Our approach allows users to write down their own visual language and removes the existing limitation of one pair of gestures per widget. The resulting parser became applicable in different contexts besides designing user interfaces. The result is a functional implementation of a parser designed to handle Relational Adjacency Grammars customized by the user..

Keywords

Calligraphic Interface, Visual Parser, Visual Syntax Parsing, Visual Grammar, Relational Adjacency Grammar, Widgets, Gestures, Sketches.

1. INTRODUCTION

Every programmer, computer engineering student, software engineer or interface designer has a natural desire to have some simple and immediate way to translate sketches and interface drafts from paper to functional source code. This concept would always apply to any given programming language they're working on in their projects. To overcome this obstacle, some researchers have been working on calligraphic interfaces for translating sketches to functional interface prototypes.

In many cases the calligraphic interfaces focus on recognizing individual strokes under an application context. In contrast, the approach in this paper presents a parser that takes the sketched shapes one by one, and uses them to recognize widgets according to sets of rules which are represented by topological relations.

2. RELATED WORK

DENIM[4] is among the first to allow the creation of interface prototypes. This was achieved by use of a pattern recognizer to identify both commands and strokes made by the user. The DENIM[4] project is also relevant because it allows multiple pages and a zooming view for the creation of storyboards for the user interface behavior. A great feature in this project was its ability to export the pages created by the user to HTML. The major limitations to DENIM[4] were the problems which were present when the pattern recognizer would not allow a sketch to be used correctly or when the recognizer would correct a stroke not meant to be corrected.

Another approach was JavaSketchit [5]. This used the CALI [3] recognizer to identify which geometrical shape was the closest match to each user drawn sketch, and used the shapes as input tokens. Each shape was crossed with all previously drawn ones to identify which pairs represent widgets and then decide which to keep in the interface prototype. A great feature was that this project would export the prototype to fully functional Java. Still, the project had limitations. These were the fact the user couldn't choose which widgets were part of the visual language, or what gesture pairs to use in each widget and which spatial relations to use.

3. THEORETICAL BACKGROUND

Grammars have been a common concept present in logic since early computation concepts and became a root topic in computational sciences. Usually grammars are described as (N, T, P, S) , a quadruple where N is the set of non-terminal symbols, T is the set of terminal symbols, P is a set of valid productions and finally S is the starting symbol. A simple example of a regular grammar definition would be the one presented in Fig. 1 where all the sets and productions are clearly defined.

This information is usually enough for most language concepts and applications, such as textual programming language parsers. However, for the approach presented here, regular grammars do not present a structure which holds enough information, in that they lack a way to specify the constraints needed to describe each non-terminal symbol in detail.

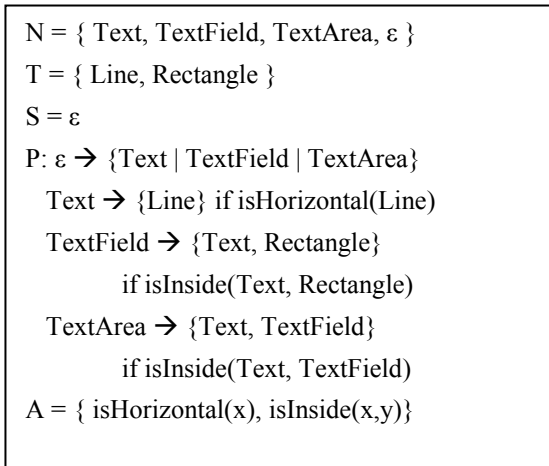


Figure 1: Example Relational Adjacency Grammar [2]

For example, visual languages are not able to describe spatial constraints such as those that indicate whether a sketch is inside another as in Fig. 3. Towards this goal, we adopted Relational Adjacency Grammars [2] (RAG). A RAG is defined as a quintuple (N, T, S, P, A) where N, T, S and P are the same as in regular grammars and A is a set of constraints which allow to keep the extra information required.

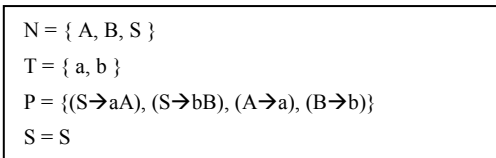


Figure 2 – Example of a regular grammar definition

In Fig. 5 we demonstrate how this information is used to describe Figs. 2 through 4. With the extra information it becomes possible to unambiguously identify each widget.

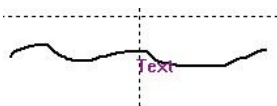


Figure 3 – A sketch representing Text.

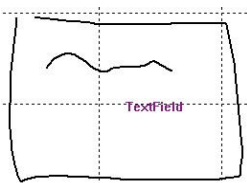


Figure 4 – Set of sketches that represent a TextField

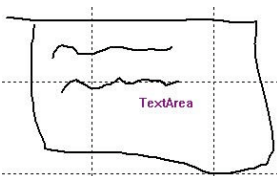


Figure 5: Set of sketches that represent a TextArea

There are modifiers other than “if” as shown in Fig. 5, though for this application only the “if” is utilized for the definition of widgets. Finally, establishing a bridge between the terms in use inside the application and the terms in the grammar:

- N represents the non-terminal symbols which are also known as widgets;
- T represents the terminal symbols which are known as gestures or sketches;
- S is the empty symbol for the reason that the templates hold no valid information until it is instantiated;
- P is the set of valid productions, which are stored as templates after read from a file;
- A is the set of rules which represents the topological relations that can be applied for the validation of widgets when new input is presented.

4. THE APPLICATION

As mentioned, this project focus on the implementation of a parser which uses a RAG to allow a better mechanism for the syntax analysis using sketched inputs, especially when there is no order when drawing the shapes. The architecture of the resulting application is shown in Fig. 6, where is visible how both the sets with non-terminal symbols (widgets) and the valid productions (templates) are supplied. The CALI [3] recognizer receives the sketched shapes and identifies them as gestures by application of fuzzy logic. As a result, the possible gesture results considered for this project are {Line, Arrow, Triangle, Rectangle, Diamond, Circle, Ellipse, Copy, Cross, Move, WavyLine}. After processing the sketch, the gesture with highest probability value is sent to the parser itself. The parser uses the gesture and tries to see if any of the temporary widgets can change its state, followed by generating new widgets from the templates. Finally, generates a list of complete widgets. These last are widgets that have gathered all the gestures they were supposed to collect, and validated all the respective rules. These rules can be either relational constraints [2] or topological relations [1,2]. The topological relations considered are either adjacency relations, overlap relations, or metric relations. These cover basic concepts where one gesture is inside another, or intersecting each other, being “to the right” of a gesture, or “to the left”, and were even considered all cardinal directions. For the metric relations were considered cases like two gestures having similar areas. In Fig. 7 is shown one example of a “grammar.xml” file where some rules are demonstrated for widget declaration.

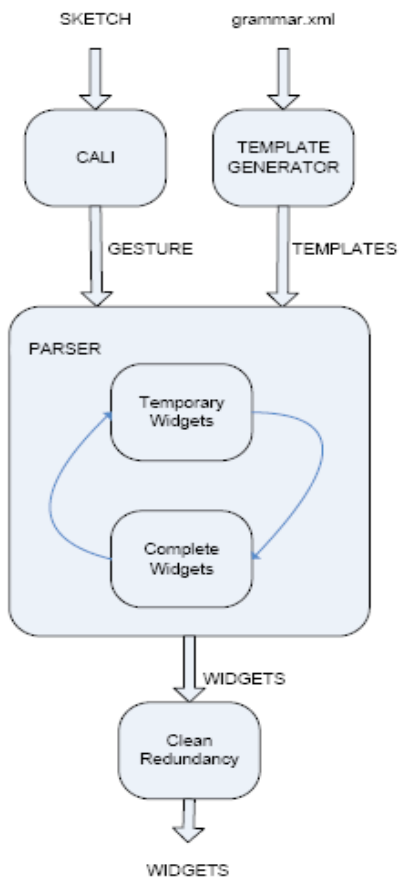


Figure 6: Architecture of the developed application

Notice still in Fig. 7 the presence of the `id` attribute which is used to assure order in how parameters are applied in rule validation. Also the distinction between `isTrue` and `isFalse` tag names for the rules allow to use the NOT operator.

To prevent false positives, routines are called to clean the global widgets lists, by removing widgets that are only parts of the larger ones, also by removing replicas and by ensuring that any gesture is only used once in the prototype.

The parser itself is based on the algorithm shown in Fig. 8, where the parsing mechanism is summarized. `ParseIncludingCompletes` takes the transformed input and uses it as input for both temporary widgets and templates. Visible in the code is that for each complete widget found, the routine will make a recursive call using that complete widget. This call allows the user to make use of a widget as piece for another widget declaration. One last feature is that each recursive call will create its own temporary widget list to avoid entering a state of infinite recursive calls.

There is one last task associated to the widget parsing, which is answering the question “which rules are needed to validate each input in a widget?”. In result, a temporary widget can be a valid hypothesis or not. This task is performed in the `addPiece` routine as shown in Fig. 9.

```

<widgets>
  <widget type="Label">
    <symbol type="Line" id="1"/>
  </widget>
  <widget type="TextBox">
    <widget type="Label" id="1"/>
    <symbol type="Rectangle" id="2"/>
    <isTrue name="isInside"
      param1="1" param2="2"/>
  </widget>
  <widget type="Button">
    <symbol type="Rectangle" id="1"/>
    <symbol type="Rectangle" id="2"/>
    <isTrue name="isInside"
      param1="1" param2="2"/>
    <isTrue name="haveSimilarAreas"
      param1="1" param2="2"/>
  </widget>
</widgets>
    
```

Figure 7: Sample contents of the “grammar.xml” file

5. RESULTS

This project has not included usability tests and inquiries since these were the object of existing results reported from the previous project (JavaSketchit). The main task consisted on converting the previous project from Scheme and C to C++. Another task was to allow users to be able to define widgets for use in the parser by means of a grammar specified in external XML files, rather than having it embedded in the code. Also, effort was spent on making the parser work in real time thus allowing users to see widgets being identified while drawing.

The results in Table 1 show that our parser is able to recognize nontrivial elements in real time, which is desirable for interactive applications. The larger values for memory consumption and parsing time are result from having ellipses sketched in the prototype. This happens because they store the greatest amount of data. Despite that, what can be observed is the average parsing time taken per gesture keeps under 0.1s, which is a reasonable value for real time interaction with the developed application.

Table 1: Average results obtained during performance tests

#Gestures per Prototype	Parsing Time per Gesture (seconds)	Memory Consumption (Byte)
9	0,00819	63.984
14	0,01061	120.432
25	0,01890	201.288
35	0,07838	411.328
43	0,05584	335.672

Engaging Virtual Agents

Ido A Iurgel
Dep. Digital Storytelling, ZGDV e.V.
Rundeturmstr. 10, 6423 Darmstadt, Germany
Ido.Iurgel@zgdv.de

Adérito F Marcos
Departamento de Sistemas de Informação, DSI
Universidade do Minho, Portugal
marcos@dsi.uminho.pt

Abstract

Embodied virtual assistants normally don't engage the user emotionally. They fulfil their functions, e.g. as shopping assistants or virtual teachers, factually and emotionless. This way, they do not explore the full potential of the presence of an embodied character. In real life, the personality of the teacher or salesperson, their ability to involve and even to entertain is essential for their success. But how much of these "soft factors" can be translated into behaviour of virtual agents? Which kinds of virtual personalities are appropriate for which group, and in which context?

We call virtual agents with engaging "soft skills" Engaging Virtual Agents. This paper presents a software platform employed for experimenting with soft skills and for creating different personalities of virtual agents. The focus of this platform is on authoring principles that facilitate the cooperation of content creators and computer scientists. We also present "Julie", an example that was shortly concluded as part of a research project commissioned by SAP AG. Julie is a virtual sales assistant that employs actively emotional expressions and narrative techniques, in order to provide additional motivation for the customer to visit and to remain at the virtual shop.

Key-words

Embodied Virtual Agent, Virtual Character, Authoring, Personality, Emotion, Virtual Assistants

1. INTRODUCTION

In daily life, interacting with a person that is fulfilling a certain duty – a functionary or employee, a sales person, a teacher, a work colleague – normally involves more than purely accomplishing rationally "hard" transactions. "Soft", emotional factors play an eminent role for the success and the stability of the processes. E.g., a teacher should expect better learning results and continuity of work if the students feel at ease with him or her; a sales person is more successful if the customer feels sympathy and laughs at the jokes of the vendor. In short, "soft", social and emotional skills matter very much, in daily life. Their importance for the efficient operating of organizations and for the accomplishment of rational tasks in a social context is well known and studied.

We employ the term "virtual assistant" in a wide sense that covers any kind of virtual functionary, e.g. teachers or sales persons. Currently available embodied virtual assistants show only very rudimentary soft skills and personality traits. E.g., a virtual shopping assistant usually displays a constant smile as the only sign of emotional involvement, remaining otherwise passive and emotionally shallow.

We think that a great potential of engaging the user is being missed, if virtual functionaries are not endowed with advanced soft skills. But what exactly are the soft skills

required for a technically feasible virtual agent? Which kind of emotional, not task oriented behaviour is useful, and what is superfluous or even disadvantageous? Certainly, R&D on social skills for virtual agents has to start with everyday phenomena of human-human interactions; but there are many reasons not to aim at an exact simulation of human soft skills.

It is important to keep in mind that a virtual agent remains a metaphor; there are certain analogies between a virtual and a human agent, but there are also essential differences. First, we won't succeed in simulating exactly human social behaviour in virtual agents, in any predictable future, because of technological limitations. E.g., virtual agents are far from being able to understand subtleties of emotional and verbal expression of a human counterpart, and thus cannot be very precise when employing social skills. We must take those technological limitations into account, when designing social skills for virtual agents; otherwise, we would postpone really building them into an indefinite future.



Figure 1: Anna, the shopping assistant of IKEA, is an example of a monotonous virtual agent (www.ikea.com)

Second, social skills for virtual agents must differ from human skills because virtual agents are perceived and evaluated differently by end users [Bente02]. For example, a charming smile of a virtual assistant won't touch a human user as deeply as the smile of a sales person; but a behaviour that would be incontestable for a real functionary, e.g. lamenting about an unhappy love affair, might be funny and entertaining when presented by a virtual agent.

Because the required social skills and personality traits of virtual agents are not yet well understood, we have build a software platform for virtual agents that facilitates the creation of different virtual agents with varying soft skills. We think that content creators require tools and methods to experiment with different ideas and intuitions, thus incrementally developing technically feasible and appealing virtual behaviours. It is not accurate to implement models derived from the analysis of human behaviour without taking into account the peculiarities of the required behaviour of virtual assistants. For example, virtual agents' "emotion psychology" might be required to be rather different from that of humans, because of the specific perception of virtual humans by end-users. Therefore, it is necessary to allow content creators to experiment with ad-hoc models and behaviour structures, and currently, the major R&D task consists in developing a framework that enables these experiments, in cooperation of content and of computer science specialists.

In this paper, we will discuss the software platform employed to experiment with social skills and personalities of virtual assistants, and we present *Julie*, a virtual shopping assistant created as an example for SAP AG. After a review of related work in section 2, we will discuss, in section 3, some important general problems of creating soft skills for virtual agents. Section 4 will expose the technical platform, with emphasis on authoring issues, and section 5 describes the exemplary implementation of *Julie*. Section 6 addresses future issues, and section 7 contains concluding remarks.

2. RELATED WORK

A. Gulz [Gulz05] provides an excellent overview and discussion of social skills in the realm of virtual assistants, especially concerning virtual tutors. Though some very sophisticated systems were developed (e.g. Bickmore [Bickmore03], Baylor and Kim [BaylorKim05]), no system was build from scratch on with emphasis on efficient authoring and on fostering experiments by content creators. Gebhard et al. [Gebhard03] have developed an authoring system for interactive applications; their solution does not integrate chatting and does not allow dismissing the edges of the directed graph they employ.

Astonishingly, none of the agents was build with the expressed intention of being entertaining and funny, and no behaviour patterns inspired by drama were employed. Nijholt [Nijholt05] has discussed humorous acts of embodied virtual agents to smoothen human-computer inter-

action problems. Most current virtual shopping assistants are commercial products, and their details are not published. However, no one seems to possess the guiding and pro-active faculties presented in this paper. The technology discussed in this paper is based on the work described in [IurgelZiegler05].

3. SOFT FACTORS FOR VIRTUAL AGENTS

Basically, we are currently "decorating" minimal, functional behaviour of virtual agents with entertaining, relaxing, or motivating additional elements. Therefore, the soft skills of the agents must be interwoven with rational and functional faculties. The concepts and technologies for interweaving assistance functions and soft factors are at the focus of the research presented in this paper. We want to provide a framework for experiments that will also help us understanding what assisting virtual characters can sensibly do, besides fulfilling their duties.

At this stage, we are interested in short remarks and short emotional dialogue games that the virtual agent can throw in whilst providing information, and in short social games that enclose the functional, assisting behaviour. I.e., we are studying how to maintain the assisting functions while at the same time adding more social and emotional factors. For example, we want a virtual shopping assistant that always will say the price of an item, when the user requests it, thus staying within its role, but that also throws in emotional remarks and interjections, and that plays engaging "mini-dramas".

Mini-dramas are temporal structures that create suspense and end with a resolution. A mini-drama is a minimally short narrative structure¹. It aims at providing additional structuring of time for the user, who should feel compelled to learn the solution. We expect mini-dramas to enhance motivation, enjoyment and fun. But we also expect them to enhance the social and emotional dimension of the communication, in an analogy to the creation of social bounds and reciprocity in real human interaction.

The suspense of a mini-drama should provide additional reasons for the customer to remain at the shopping site, or for the pupil to continue his e-learning session. E.g., Julie (cf. bellow) will begin throwing in unhappy and nasty remarks which she² first refuses to comment (conflict), increase their intensity until a climax is reached, and then confess that she desires to become friend with the user, but doubts whether this wish is reciprocal (resolution). We expect the users to be pleased and laugh at this turn around, and not to be deeply touched by this virtual advance. Following drama theory [McKee97], this mini-drama has revealed a hidden, unexpected trait of Julie's

¹ A mini-drama is roughly equivalent to what Mateas and Stern [MateasStern05] call a beat, following the terminology of McKee [McKee97] though the term beat wouldn't be appropriate here because a mini-drama is not necessarily embedded within a larger narrative structure, but can stand on its own.

² We think that is sounds more natural to employ the personal pronoun "she", instead of "it", when referring to the female name "Julie".

“personality”, namely her (alleged) affection towards the user; this uncovering of a “hidden”, unexpected layer is exactly the main element of a good story. As in cliff-hangers of TV-soaps, mini-dramas could also be used to create continuity, suspense and curiosity when returning to a learning session or to an internet shopping site.

Developing mini-dramas require much authoring efforts. Automated creation of mini-dramas out of general and generative drama rules is out of current technology’s reach. This means that only a limited set of mini-dramas will be available, for any application. This poses challenges to commercial exploitation; there, a constant production of additional mini-dramas for regular customers might be necessary.

It is not common for real sales persons or teachers to actively employ mini-dramas to engage their counterpart. But this is an example of the major difference between virtual and real agents, and clarifies why we need a framework for experimenting with the specifics of virtual assistants: A real teacher could lose his/her face and privacy, when uncovering hidden personality layers and feelings; a real sales person could strategically flirt with a customer, but would appear aggressive and lose credibility when playing with deep feelings. Those are no concerns, when it comes to a virtual teacher or shopping assistant.

The really challenging interaction design questions in the virtual realm are related to balancing and estimation of effect. “Balancing” means that, in applications that employ virtual assistants, some hard criterion for success exist and that extra-functional behaviour of the virtual agent must, at the end of the day, have contributed to this goal. As in any real working context, we are better off with social skills, but we won’t achieve the targets if we only focus on socializing. Some difficult balancing is required. For example, the funny behaviour of a virtual shopping assistant must not demand the whole attention of the customer, but should be useful in attracting more site-visitors and motivate them to buy. A virtual tutor in a business context must not consummate too much time with the creation of emotions, because the emotions must be beneficial for the final learning success.

“Effect” is a problematic design aspect because the impact on a particular user is still very difficult to foresee. Here, again, the analogy between a virtual and a real human can be misleading. If the user laughs at a joke told by a virtual character, how much is he/she laughing about the joke, and how much about the fact that a soulless thing is pretending to tell a joke, employing a dry synthetic voice and imprecise mimic? In the later case, would the user still laugh at the n-th joke? The effect of a behaviour well known on humans is not easy to foresee, when simulated by a virtual human.

Though balance and effect can be evaluated formally, the process of optimising them is essentially an intuitive, experience driven, and even artistic process, in the same vein as screenwriting. Hence the importance of a system

that supports experiments and that can be authored by content creators that are not computer specialists.

4. THE SOFTWARE PLATFORM

4.1 Implementation

The runtime environment is a client-server system for the internet which supports virtual characters that are

- proactive
- assisting and guiding
- able to throw in sensibly remarks, jokes, teasing, and other interjections
- able to play mini-dramas
- responsive to mouse click and able to chat
- flexible and easy to author

The server and the authoring tool were written in Java, the communication employs AJAX³. The complete logic is on server side; on the client side, we use the internet explorer with the Haptex⁴ plug-in for virtual character rendering, which employs the speech synthesis of the Windows client through the Microsoft SAPI 5.1 API. The authoring tool *Cyranus* was developed at ZGDV e.V. (cf. [Iurgel06]), using Java Swing.

4.2 The Quest for Accessible Authoring

Accessibility of authoring is an essential requirement for the envisaged target of intuitively experimenting with possible social skills for virtual assisting agents. Content creators with some computer affinity, but who are not computer scientists, must be able to use the tools. Equally, the authoring system must be easily adaptable and customizable by computer scientists, in dialogue with the content creators.

The most established principles for authoring of interactive applications are some variation of a directed graph. They are very intuitive when the graphs are not too dense, the guard conditions not complicated, and the amount of possible paths not too large. In the scenarios we envisage, there are many situations in which directed graphs can be used. Thus, an important feature of our approach consists in enabling the use of directed graphs wherever the content creator deems it appropriated, and of other control strategies when necessary. There must be a smooth transition from the use of directed graphs to those other technologies, and the author must be able to begin with graphs, or even with a linear sketch, and be able to later replace the sketch easily, when the graph has become too complex.

³ “Asynchronous JavaScript Technology and XML”, a technology that extends the use of Javascript within web applications.

⁴ Cf. <http://www.haptex.com/>

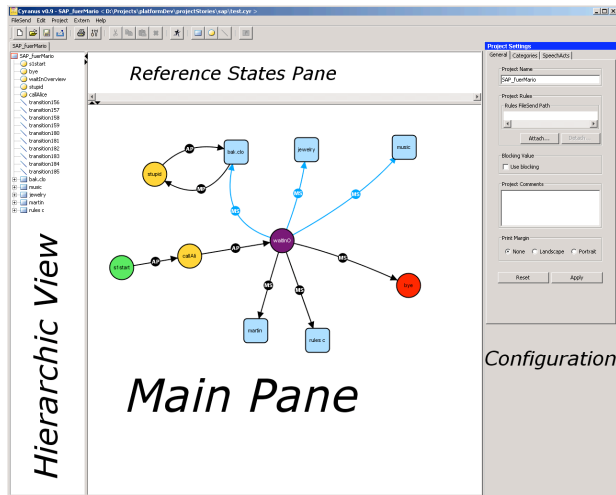


Figure 2: Screenshot of *Cyranus*. On the left a tree view of the hierarchy, on the right the configuration panels

Short dialogue games are an example of the sensible use of directed graphs within an interactive application with natural language (typed) input. Of course, only short pieces of the dialogue are predictable enough to be handled by graphs.

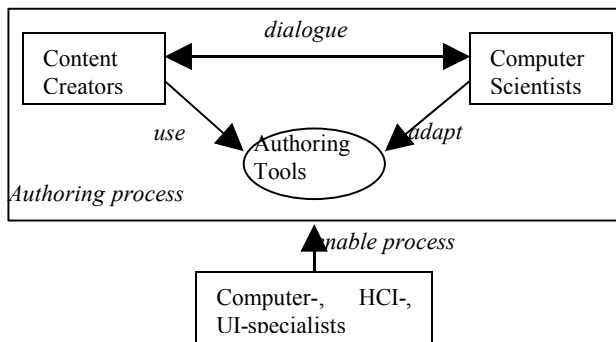


Figure 3: The process of adaptation of the authoring tools by computer specialists must be already addressed when designing the authoring tools.

For these cases we allow, at any time, for the use of a hierarchic transition network, inspired by Harel’s State Charts [Harel88]. Figure 2 is a screenshot of the authoring tool *Cyranus*. The figure depicts the directed graph representation, derived from State Charts. Composite states are represented by squares; double-clicking on a square will show the states that are encapsulated by the composite state, i.e. the state that are “inside the square”. Terminal states are circles; circles do not contain other states inside.

It is well known that directed graphs cannot be employed universally; for example, they are inappropriate if a large number of permutations over a set of states is possible. Figure 4 shows an already forbiddingly complex graph over a set of only six states of *Cyranus*; the most complex guard conditions required to control the flux are not shown. Because of this complexity, other “next state” activation procedures must be devised.

Other control strategies than the use of directed graphs are integrated by a delegation chain: An external event that does not lead to the activation of a state within the transition network framework is passed over sequentially to other so-called “activation engines”, which in their turn could activate a state, in this case breaking the chain of delegation. The engines are also permitted to fire without the occurrence of an event. Every activation engine is attached to a single composite state, through meta-data of this state, and it may activate only terminal or composite states that are located within this composite state. Thus, encapsulation is fully preserved.

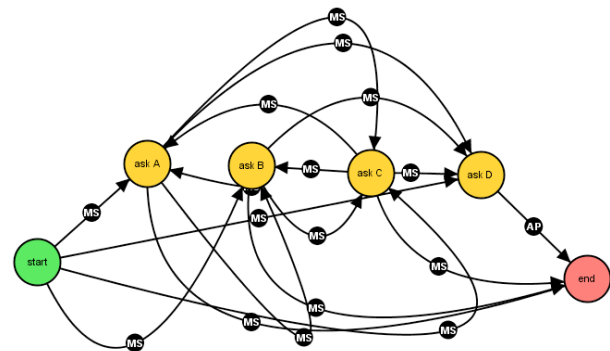


Figure 4: A scenario where different inputs may cause many permutations over a set of states

We currently employ two activation engines: The production system Jess⁵ (we are up to now employing solely the forward chaining rules), and a similarity based engine that shall implement principles of CBR (Case-Based Reasoning), which is currently being developed within the German *Virtual Human* project⁶. As a fall back engine, we also employ the chatterbot *Alice*⁷. This is not an activation engine, because it does not activate a state that is within reach of all activation engines. Instead, the *Alice Engine* generates autonomously an action that is send to the client. *Alice* is employed here because of the huge data bases that can be freely downloaded; this move extends the chatting capabilities of the virtual agents. Figure 6 represents the delegation chain; in the example of this figure, the CBR Engine (Case-Based Reasoning) only is able to handle the external event, by activating a state of the composite state this engine is attached to.

Any state that the engines of the activation chain may activate is represented by *Cyranus* as either a square or a circle. I.e. every state that could be activated is visually represented, thus guaranteeing a high degree of visual control of the possible actions of the system (with the exception of the fallback engine for ALICE, cf. above).

⁵ Cf. <http://herzberg.ca.sandia.gov/jess/>

⁶ Cf. www.virtual-human.org

⁷ Cf. www.alicebot.org

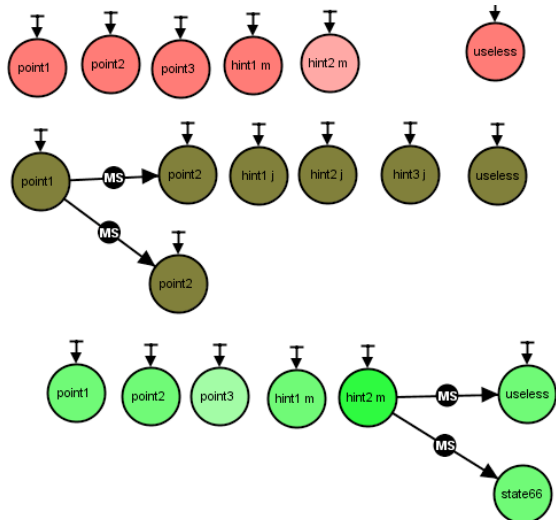


Figure 5 : Employing the transition network together with activation engines

States can be activated by activation engines because they carry meta-data. We expect content creators to be able to use the directed graphs, with the help of simplifying GUIs for specifying guard conditions, and to describe the meta-data of the states that an activation engine shall be able to activate, equally employing GUIs. We normally do not expect an author do express formal rules or new similarity functions. This is the work of computer scientists, which will work in dialogue with the content experts.

Figure 5 depicts the combined use of activation engines and of the transition network. Note that most of the circles – terminal states – are not attached to transitions, and that they have an arrow on top. This arrow symbolizes that they contain metadata, and can thus be activated by an activation engine. Note also that it is possible to combine transitions with the activation engine. Only a single state can be active at a given moment, parallelised activation is not yet being supported. If an arrow connects the currently active state to another state, in the visual representation of *Cyranus*, the system will first check whether this transition is passable; in case that the guard conditions do not hold (transition not passable), the delegation chain (cf. below, and figure 6) will result in the calling of subsequent engines, which may activate any state which contains appropriate meta-data – on the representation of *Cyranus*, this is any state (square or circle) that shows a short symbolic arrow on top. The activation engines are adapted by computer specialists for producing the required behaviour, and a content specialist then only has to add states and employ the meta-data to take profit of the engines, without knowledge of any implementation detail.

Once a rule set or function is created, it can be reused with new content. This dedicated process of adaptation and enhancement (fig. 3) is an important new feature of the authoring system: We should not expect work on the authoring tools to be concluded, at the beginning of a pro-

duction, since content creators might always come out with new ideas and requirements; the tools therefore should provide dedicated extension procedures that efficiently end up with the required features, whilst still remaining accessible for the content creators. This considerable extends the reach of the authoring process.

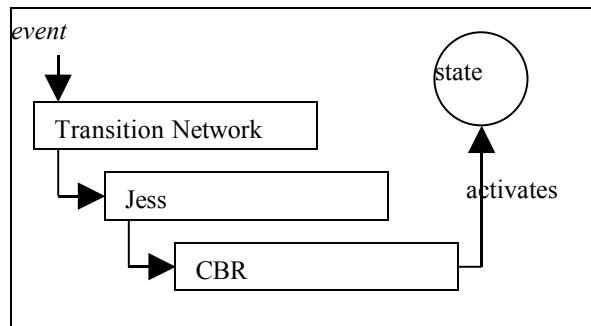


Figure 6: The chain of delegation of events to different engines. In this example, the CBR-engine responds

Thus, our authoring tool *Cyranus* also offers neat APIs and configuration for adapting the GUIs that are used to create meta-data, guard conditions, and action expressions. Internally (i.e., not visible for the content specialist), they are described with generically expressive XML (e.g., the guard-conditions are kept internally as XEXPR⁸, a very powerful, general, functional programming language in XML). For example, figure 7 shows partially the GUI for scripting utterances and other behaviours of the virtual agents. This exchangeable scripting GUI is integrated within *Cyranus*, and it outputs XML to the main *Cyranus* framework, which integrates this XML into possibly more complex XEXPR expressions. The non-expert content creator will not be involved into any XML-coding issues.

5. JULIE – AN EXAMPLE OF AN ENGAGING VIRTUAL AGENT

We’ve created an exemplary virtual shopping assistant for SAP AG, in order to study both the creation process of virtual assistants and certain non-standard behaviour patterns for virtual assistants.

⁸ Cf. <http://www.w3.org/TR/xexpr/>

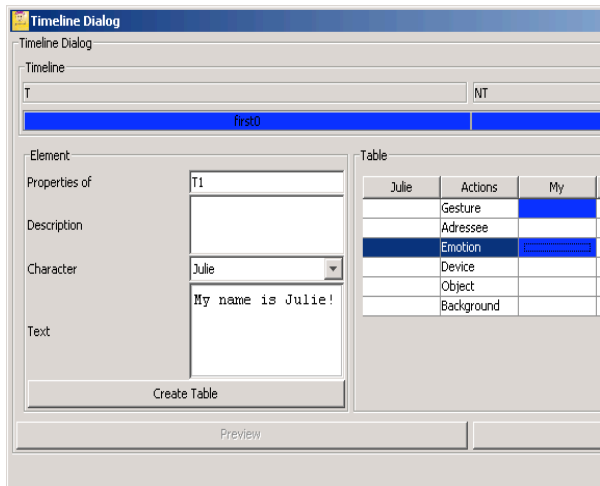


Figure 7: Part of the GUI for scripting behaviours for virtual agents

5.1 Julie's Concept

The exemplary cloths shopping site is intended for young women and girls. We expect that a considerable part of this group will react to attributes as “cool”, “trendy”, “funny”. The visual design of the site is inspired by real high class cloth shops, in order to create a young, sophisticated and demanding atmosphere that differs from usual internet shopping sites. Figure 8 exemplifies the design of the shopping site. It depicts the main page of the cloth section. The user clicks on one of the cloths displayed on the right, and the item moves forward, as in a catwalk. The user may chat with Julie at any time, using the chat-box located below Julie.

We wanted the shopping assistant *Julie* to be able to guide the customer through her shopping experience. Julie should comment on mouse clicks of the user, e.g. saying “If I were not caged in this computer, I would myself by this jacket!”, or initiating short dialogue games, e.g. “do you want me to tell what I really think about this cloth?” – “yes, please” (typed) – “it will make you look overweight”. Julie should not be repetitive when the user clicks more than once on an item, and should actively guide the user to items not yet examined, e.g. saying “look at the jacket on the top right corner. It will certainly fit you!” – (user clicks at item) – “And? What do you think? Am I not right?”

Julie's remarks are not only informative, but sometimes teasing, joking or even bitchy. Of course, this is intended to create fun. Julie is pro-active, and will e.g. sometimes try to instigate some off-topic chat, e.g. on the party of last night. Julie also can comment on the overall session and the reactivity of the user, e.g. saying “you are wasting my time if you do not do anything”. At any time, the user can freely click on the items and chat with Julie.

These examples already indicate the personality profile we have created: Julie is much more active, demanding, direct, and even nasty and aggressive than other virtual assistants.

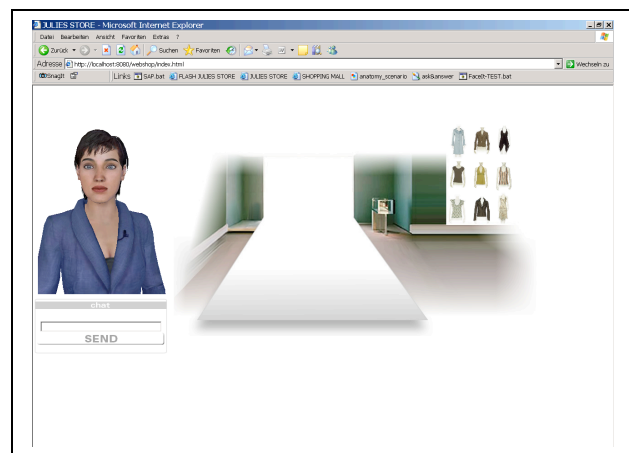


Figure 8: The design of the shopping site for Julie

Julie is intended to be a rather extreme example of a personality rich, active, emotional assisting agent. We want to understand whether and by whom this is perceived as distracting and offensive, or funny and attracting. For commercial use, marketing considerations would certainly require the reduction of some parameters.

We are also experimenting with mini-dramas for Julie. Currently, we have implemented the mini-drama “I want to be your friend”. Julie will begin by throwing in some nasty remarks like

“I don't think that we are going to get along with each other” – “why?” (typed) – “I do not want to tell”.

A bit later, Julie will resume the topic:

“Do you really want me to tell you what is the problem between us?” – “yes” (typed) – “I think I really want to be your friend, but this is not possible...” (makes a very sad face).

This mini-drama shall, at its beginning, create suspense, because the user shall be intrigued to know Julie's reasons. It shall after its resolution initiate some more intimate exchange, and the creation of some sort of social bounds between user and Julie.

5.2 Technological Issues

The guiding behaviour is generated by a rule-set expressed in Jess. Figure 9 is a short example of a rule written in Jess. Jess activates a corresponding state, if the user

clicks on or mentions a piece of cloth, and activates states that guide the user to items not yet examined.

```
(defrule updateNextPoint "keeps priority for click
states up to date"
  (declare (salience 100))
  (state (id ?id) (visits ?v))
  (meta (state ?id) (name hotspot) (value ?
region))
  (meta (state ?id) (name priority) (value ?p))
  (meta (state ?id) (name type) (value point))
  (start (id ?id) (visits ?v))

  ?next <- (next (hotspot ?region) (type point)
(priority ?oldp&:(< ?p ?oldp)))
```

Figure 9: Example of a Jess-rule

The states carry meta-data as

```
<ruleData hotspot="partyDress" priority="5"
type="point" />
```

indicating that this state shall be activated by the Jess-engine whenever the user clicks for the fifth time on the party dress item.

The meta-data

```
<ruleData hotspot="partyDress" priority="1"
type="hint" />
```

for example, means that this state will be activated only if the party dress is not being noticed at all, and that the state contains behaviour that will guide the user to look at this dress.

States can also be activated by typed user input. For the resolution of the input, we employ either a regular expression, or reuse the so called “reduction” mechanism provide by *Alice*. E.g., the user input

“do you love the grey coat?”

will be changed internally into

“do you like the grey coat?”.

This can be matched against the meta-data

```
<ruleData hotspot="partyDress"
speechact="do you like the grey coat?" pri-
ority="1" type="chat" />
```

and then the state can be activated. Note that a short dialogue game, authored with directed graphs, could follow:

“do you like the grey coat?” (typed) – “yes, very much. What’s about you?” – “I don’t know” (typed) – “if I were you, I would buy it!”

The guard conditions for the transitions of this short dialogue game can contain either regular expressions to catch up predictable input, represented internally for example as

```
(\w|\s|\d)*yes(\w|\s|\d)*
```

for any sentence containing “yes”, or again recur to the reduction functions of *Alice*.

Actions of states are mostly dialogue acts expressed in XML, which are send as direction to the client. The

format is derived from RRL [Piwek02]. For example, the sentence “don’t you feel good?” will be coded as

```
<dialogAct>
  <sentence id="s_1147866678280">
    <word id="w0">Don't</word>
    <word id="w1">you</word>
    <word id="w2">feel</word>
    <word id="w3">good</word>
    <textSign id="ts0">?</textSign>
  </sentence>
</dialogAct>
```

5.3 First Experiences

We have not made yet a formal evaluation of *Julie*. Our informally acquired feedback about Julie’s behaviour ranges between extremes. Seldom had we observed indifference. For example, a middle aged woman without computer science background felt sincerely repudiated and preferred not to be confronted twice to Julie. But more often, guests and occasional users laugh out loudly at Julie, and immediately start explorations of the system, in expectation for the next funny thing to happen. Experiences with the reaction of a group of young girls of the target age, visiting the German institute, indicated that we should be on the right track.

Concerning the authoring process, the experiences are very favourable. For example, many of the funny remarks and dialogue games of Julie were made by a 16-year old pupil without any deeper computer background who was making a short internship, and the total amount of time required for instructing him did not exceed some very few hours. *Cyranus* has proven to be flexible and powerful enough to be employed in practice, without structural changes, within three other projects.

We initially underestimated the learning curve of the rule-system Jess. It seems now out of reach that someone without a computer science background should extend the rules of the system, which we at the beginning thought would be possible.

6. FUTURE ISSUES

We are currently implementing a second experimental scenario for Julie as a kiosk within a shopping mall. The distinctive feature of this variation is the video-based emotion recognition of the user, and the integration of this input into the behaviour of Julie. In this scenario, Julie will explain the way to different shops of the mall, but will also, from time to time, tell a joke. We need the emotion recognition to detect whether the visitor enjoys Julie’s attempts at being funny. This is likely to vary very much among visitors. The frequency of entertaining remarks will depend on the recognized emotion. In this scenario, adaptation of the system to the user is important because the visitors cannot be expected to configure or to adapt to the system.

Within scenarios of longer duration, e.g. tutoring systems or certain internet shopping sites, it is certainly more im-

portant that the user is enabled to adapt the system to his/her preferences by choosing a preferred virtual character, than creating a virtual agent that would try to adapt to the very different possible user types.

We have not yet made any formal evaluation of the “Julie”-system, since we are currently still at the phase of acquiring experiences and fine-tuning content. In fact, a formal evaluation of Julie would certainly prove to be a most difficult endeavour. The evaluation of Julie would be a success if a shopping site sells more because of its social and emotional behaviour. This might be the case even if a majority of potential customers strongly dislike Julie, as long as a relevant minority enjoys this assistant.

Most probably, Julie will remain, for us, an in-between step useful for acquiring experiences, propagating ideas and fine-tuning technology. Only the next assistant is going to be formally evaluated, which is likely to be a virtual tutor, because the learning effects of a virtual tutor are easier to evaluate. In a former evaluation of a learning system that employs virtual characters⁹, we have found some positive effects, and indicators of “pleasure” were significantly high. But, perhaps astonishingly, the pleasure was more associated with “relaxation” than with “excitement”, which goes along with other findings about the fear reducing effects of virtual characters [Bente02]. An important question for future evaluation of our system will be whether it will be possible to maintain or increase positive effects, while moving the “pleasure” indicator to the side of “excitement”. This is presumably important for example when younger user groups shall be addressed.

A major issue that is being tackled concerns augmenting the generativeness of the system. E.g. Julie employs shallow natural language processing and shallow emotional behaviour. I.e., no generic modules are employed that would allow Julie to generate dialogue acts and dialogue games out of language independent data bases and context information, and no emotional models control Julie’s emotions. They are scripted, and unforeseen sequences of scripts may cause weird emotional behaviour. It was necessary to start shallowly because generated behaviour is not entertaining and witty. But on the long run, a combination of both scripted and modelled behaviours will be necessary, for the sake of modularity, parameterisation, and diminution of the authoring expenses.

7. CONCLUSION

We think that virtual embodied agents possess important potentials for motivating, relaxing, or entertaining users, when they present some social skills which they can employ without neglecting their proper functions as virtual teachers, sales persons, or servants. Following Gulz, we do not expect any simple, generally valid findings. Some people are likely to adore some kinds of virtual characters, others will hate them, and some persons will prefer to dis-

pense with virtual agents altogether. The social preferences and needs for real or virtual company are certainly very different and individual. Engaging virtual agents also introduce a role play and game element into possibly serious contexts that won’t be appreciated by everyone.

But nevertheless, many people are likely to profit from engaging virtual assistants, to enjoy them and to experience pleasure at their presence. The current practice of creating neutral, inoffensive virtual assistants with little personality is probably missing the point, because differences between preferences of different persons are being neglected, and, at the end, only very few persons really end up enjoying the virtual agent. A virtual personality that is offensive for a certain group can be very funny for another; an agent that evokes protecting and nurturing instincts (reminiscent of the Tamagotchi) won’t captivate everybody, but certainly a relevant minority. We think that the solution lies in learning to create different, pronounced personalities of virtual assistants that will please certain groups, and to let the persons choose the character, if any, that is most appropriate.

The software platform that we’ve presented enables the creation of different virtual personalities with different social skills, in cooperation of content creators and computer scientists, and to experiment with these agents. *Julie*, the virtual character that we have presented as an example is rather active, and a bit nasty and aggressive, and employs emotionally loaded mini-dramas to catch the attention of the user. Many people enjoy and laugh at this.

REFERENCES

- [BaylorKim05] A.L. Baylor, Y. Kim. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education* 15, 1-18.
- [Bente02] G. Bente, N. C. Krämer. *Virtual gestures: Analyzing social presence effects of computer-mediated and computer-generated nonverbal behaviour*. Paper presented at Fifth Annual International Workshop Presence 2002, 8.-10. October 2002, Porto, Portugal.
- [Bickmore05] T. Bickmore. Relational agents: effecting change through human-computer relationships. PhD Thesis, MIT.
- [Gebhard03] P. Gebhard, M. Kipp, M. Klesen, T. Rist. *Authoring Scenes for Adaptive, Interactive Performances*. In: Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS’03), Melbourne.
- [Gulz05] A. Gulz. *Social enrichment by virtual characters – differential benefits*. *Journal of Computer Assisted Learning* 21, 405-418.
- [Harel88] D. Harel. *On visual formalisms*. *Communications of the ACM*. V. 31 15 514-530.
- [IurgelZiegler05] I. Iurgel, and M. Ziegler. *Ask & Answer: An Educational Game where it Pays to Endear your Capricious Virtual Character*. IVA 05, Interna-

⁹ This yet unpublished evaluation was conducted within the art-E-fact-project, cf. www.art-e-fact.org, IST-2001-37924. More information on this evaluation’s results can be obtained from the authors upon request.

- tional Symposium on Intelligent Virtual Agents, September 12-14, Kos, Greece.
- [Iurgel06] I. A. Iurgel. *Cyranus - An Authoring Tool for Interactive Edutainment Applications*. In: Lecture Notes in Computer Science (LNCS), Springer, S. No. 3942 (2006), p. 577-580.
- [MateasStern05] M. Mateas and A. Stern. *Procedural Authorship: A Case-Study of the Interactive Drama Façade*. Digital Arts and Culture (DAC), Copenhagen, November.
- [McKee97] R. McKee. *Story: Structure, Style, and The Principles of Screenwriting*. Regan Books.
- [Nijholt05] A. Nijholt. *Conversational agents, humorous act construction, and social intelligence*. In K. Dautenhahn (Ed.), Proceedings AISB 2005: Social Intelligence and Interaction in Animals, Robots and Agents. Symposium on Conversational Informatics for Supporting Social Intelligence and Interaction, 1-8. Hatfield, England: University of Hertfordshire.
- [Piwek02] Piwek, P., Krenn, B. Schröder, M. Grice, M., Baumann, S. Pirker, H. *RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA*. Proceedings of the AAMAS workshop on "Embodied conversational agents - let's specify and evaluate them!", Bologna, Italy.

Algoritmo de Visão para Humanos Virtuais

Pedro Semião, Maria Beatriz Carmo, Ana Paula Cláudio

Faculdade de Ciências da Universidade de Lisboa
i21197@alunos.di.fc.ul.pt, {bc, apc}@di.fc.ul.pt

Sumário

A plataforma IViHumans destina-se a suportar a geração de cenas animadas com humanos virtuais inteligentes. Esta plataforma é constituída por dois módulos independentes que comunicam entre si: a componente do processamento gráfico e a componente do processamento inteligente. Neste artigo apresenta-se um algoritmo de visão sintética, baseado em ray casting, integrado nesta plataforma. Através deste algoritmo um humano virtual pode reconhecer o ambiente que o rodeia permitindo a interacção com os elementos da cena.

Palavras Chave

Humanos virtuais, visão sintética, ray casting

1. INTRODUÇÃO

A construção de ambientes virtuais onde se movimentam humanos virtuais é um tema de estudo com aplicação em várias áreas: entretenimento, educação, treino, quer em situações de emergência, quer em aplicações militares ou em ambientes industriais, reconstrução de ambientes históricos, aplicações médicas, tratamento de fobias, planeamento urbanístico entre outras.

Para gerar um agente virtual, além de uma aparência física realista, é necessário dotá-lo também de uma personalidade própria capaz de exprimir emoções e interagir com o meio que o rodeia. Um aspecto chave para a concretização da interacção com o meio é a incorporação de mecanismos que permitam captar o mundo envolvente, simulando a forma como os humanos apreendem o meio ambiente. Embora já haja estudos que se debruçam sobre o desenvolvimento de sensores tácteis e auditivos [Noser95], os sensores de visão são os mais desenvolvidos. Isto porque, por um lado, requerem, normalmente, equipamento menos sofisticado, por outro lado, extraem informação mais completa sobre os objectos que rodeiam o agente.

Neste artigo descreve-se um mecanismo de visão integrado na plataforma IViHumans (**I**ntelligent **V**irtual **H**umans) desenvolvida para suportar a geração de cenas animadas com agentes inteligentes [Silvestre05] [Carmo05]. Para simplificar a linguagem, designaremos no texto humanos virtuais inteligentes por agentes.

Há diferentes abordagens para a concretização de algoritmos de visão. Na secção 2, identificam-se os tipos de soluções encontradas na literatura. Na secção 3, descreve-se a plataforma IViHumans onde o algoritmo desenvolvido se enquadra. Na secção 4, apresenta-se o algoritmo implementado. Finalmente na secção 5 conclui-se descrevendo o trabalho em curso e o trabalho futuro.

2. MECANISMOS DE VISÃO

Analisando os algoritmos de visão desenvolvidos, identificam-se dois tipos principais que podem ser designados por omniscientes (*sensory omniscience*) e honestos (*sensory honesty*)

Os algoritmos designados por omniscientes têm acesso directo à descrição da cena. Assim o agente tem informação sobre todos os objectos, incluindo aqueles que não consegue ver, por exemplo, os que estão atrás de uma parede. Apesar deste tipo de visão ser mais fácil de concretizar e ser mais eficiente em termos de tempo, é mais susceptível de produzir resultados irrealistas.

Os algoritmos honestos [Burke01] tentam obter apenas a informação que está no campo de visão do agente. Deste modo, o agente só reage perante os objectos que estão realmente visíveis, atingindo-se uma interacção mais realista. Esta categoria de algoritmos pode subdividir-se em dois grupos: visão artificial e visão sintética.

Designam-se por algoritmos de visão artificial aqueles que se baseiam no reconhecimento de imagens [Noser95]. Neste caso, o agente conhece apenas a informação extraída a partir de imagens 2D. Este processo requer a utilização de algoritmos complexos e que exigem, normalmente, muito tempo de processamento. Esta técnica é sobretudo utilizada na robótica em situações em se pretende reconhecer um ambiente real à custa de imagens capturadas por uma câmara.

O conceito de visão sintética foi introduzido por Renault et al. [Renault90], correspondendo à simulação da visão do agente através de uma câmara localizada ao nível dos seus olhos. A partir da geração da imagem, usando o ponto de vista do agente, obtêm-se os objectos visíveis. Além disso, acedendo à descrição da cena, é possível consultar, nomeadamente, a posição dos objectos e informação semântica disponível. Deste modo, obtêm-se

mais informação sobre o ambiente e de forma mais rápida [Peters02].

Para obter informação a partir da imagem gerada, Renault et al. [Renault90] propuseram uma técnica baseada em cores falsas. Isto é, os objectos da cena não são desenhados com a cor definida na descrição da cena, mas usando uma cor que identifica univocamente o objecto. O resultado da geração da cena segundo o ponto de vista do agente é guardado numa matriz bidimensional que contém em cada posição informação sobre o valor do *pixel*, a identificação do objecto a que corresponde o valor do *pixel* e a sua distância aos olhos do agente. A dimensão da matriz tem de ser escolhida de modo a identificar os objectos com uma precisão razoável, mas sem sobrecarregar muito o sistema. Renault et al. usaram uma matriz 30x30 para aplicar esta técnica à detecção de obstáculos num corredor. Noser et al [Noser95] usaram uma matriz 50x50 e estenderam esta técnica juntando um modelo de memória visual baseado numa *octree*. Kuffner e Latombe [Kuffner99] aplicaram esta técnica adoptando uma matriz de 200x200 *pixels*. Neste caso considerou-se que o cenário era composto por elementos de pequenas dimensões. Objectos maiores como, por exemplo, paredes foram subdivididos em vários objectos. A cada objecto foi atribuído um identificador único. Neste caso, a memória visual era simulada através de uma lista que guardava os identificadores dos objectos visíveis.

Peters e Sullivan [Peters02] estenderam a técnica de cores falsas definindo modos de visão diferentes, recorrendo a paletes de cores distintas: modo individual (*distinct mode*) e modo em grupo (*grouped mode*). No modo individual, a cada objecto é atribuída uma cor. No modo em grupo, a mesma cor é partilhada por um grupo de objectos que satisfazem um critério previamente estabelecido. Este último modo é menos preciso porque basta que um dos elementos do grupo esteja visível para que o grupo seja assinalado como visível. Estes autores consideraram uma matriz de 128x128 *pixels*.

Terzopoulos et al. [Terzopoulos96] adoptaram um sensor de visão cobrindo um ângulo de 300° para simular a visão de peixes. A visão de cada peixe é obtida através de quatro câmaras virtuais que geram imagens com diferentes resoluções e campos de visão. Cada peixe é capaz de reconhecer um conjunto de modelos de cor de objectos em que está interessado.

Inspirando-se em modelos sobre o movimento de abelhas, Blumberg [Blumberg96] propôs um modelo de visão sintética baseado no equilíbrio da energia do movimento. A cena é gerada do ponto de vista do agente e o valor de cada *pixel* em *frames* sucessivas é usado para determinar a energia do movimento, bem como outras características do ambiente.

Uma técnica diferente de visão sintética, baseada em lançamento de raios (*ray casting*), é apresentada por Vosinakis e Panayiotopoulos [Vosinakis05]. Um conjunto de raios é lançado para a cena dentro do campo de visão do agente. Para cada raio determina-se a posição, tamanho e tipo dos objectos que são intersectados pelos raios.

O algoritmo de visão descrito na secção 4 baseia-se no lançamento de raios.

3. A PLATAFORMA IVIHUMANS

Para gerar humanos virtuais com aparência e comportamento credíveis, além de reproduzir o aspecto humano de forma realista, é necessário criar mecanismos para simular o comportamento humano. Para atingir este objectivo, a plataforma IViHumans tem duas componentes fundamentais: a componente gráfica e a de inteligência artificial. A componente gráfica suporta a modelação e animação quer do ambiente quer dos humanos virtuais. A componente de inteligência artificial é responsável pelo comportamento do humano virtual.

Uma vez que o processo de desenvolvimento de ferramentas gráficas e de inteligência artificial é normalmente moroso e exigente do ponto de vista de recursos humanos, optou-se por incorporar aplicações já existentes e de domínio público. Para a componente de processamento gráfico é necessário considerar as capacidades de modelação e de geração de imagens em tempo-real, bem como, a simulação do comportamento físico dos objectos da cena. Para preencher estes requisitos foram seleccionadas as seguintes aplicações: o Blender [hB] para a modelação 3D, o OGRE [hOG] como motor gráfico e o ODE [hOD] como motor físico. Na realidade a modelação pode ser efectuada por qualquer aplicação que exporte o modelo construído num formato que possa ser lido pelo motor gráfico (OGRE).

Para a componente de inteligência artificial foi adoptada a bancada de agentes JADE (Java Agent Development Framework) [hJ].

Como as aplicações gráficas adoptadas estão implementadas em C++ e o JADE está implementado em Java, a comunicação entre as duas componentes é feita através de *sockets*.

Uma das características principais desta plataforma é a sua modularidade. A integração de um módulo para a simulação de visão é uma peça essencial para a interacção dos agentes entre si e com o ambiente que os rodeia.

4. ALGORITMO DE VISÃO

Considerou-se que seria mais adequado incorporar na plataforma um algoritmo de visão do tipo honesto. Ou seja, um agente vê apenas os objectos que estão no seu campo de visão e identifica-os através da imagem que é gerada a partir de uma câmara colocada ao nível dos seus olhos. Uma forma de garantir que todos os objectos são detectados consiste em varrer todos os *pixels* da imagem e identificar os objectos representados. Mas esta aproximação seria muito lenta. Nos exemplos apresentados por outros autores, o maior número de *pixels* tratados foi 200x200. Podemos considerar como razoável estabelecer um intervalo de tempo dentro do qual um agente deve tomar conhecimento da entrada de um objecto no seu campo de visão. Por exemplo, considerando um segundo como intervalo de tempo razoável para detectar um objecto, o varrimento da

imagem pode ser efectuado ao longo das *frames* que são geradas durante um segundo. Se forem geradas X *frames* por segundo (FPS), o varrimento da imagem é feito em X *frames*. No entanto, se o número de *frames* por segundo diminuir, pode haver objectos que não são detectados no tempo pretendido. Para minorar este problema, o algoritmo de visão deve adaptar o varrimento da imagem de acordo com o número de FPS para manter um bom desempenho.

No motor gráfico escolhido para a plataforma havia duas formas de satisfazer os objectivos acima descritos: consultar o buffer de profundidades (Z-Buffer) ou lançar raios para a cena. Uma vez que a primeira solução ficaria muito dependente do hardware utilizado, entendeu-se que a segunda abordagem seria melhor. Com o lançamento de raios a partir da posição dos olhos do agente para pontos do *viewport* é possível identificar os objectos que são intersectados pelos raios.

Após a realização de alguns testes, verificou-se que o motor gráfico (OGRE) não fazia intersecções precisas entre o raio e objecto. O OGRE faz a intersecção do raio com a caixa envolvente do objecto (*bounding box*). Este volume é normalmente muito maior que o objecto o que conduz a situações de erro: a caixa envolvente pode ser intersectada, mas o objecto não ser efectivamente intersectado pelo raio. Se este objecto estiver à frente de outro, gera-se uma situação de falsa oclusão. Para obter intersecções mais precisas, adoptou-se a biblioteca OpCode [hOC] Optimized Collision Detection, que permite obter caixas envolventes muito próximas da forma do objecto. Utilizou-se a biblioteca OgreOpCode [hOGC] para estabelecer a ligação entre o OGRE e o OpCode.

Resolvidos os problemas de detecção correcta de intersecções, foi necessário definir quantos raios lançar, uma vez que não é aceitável em tempo-real lançar raios para todos *pixels* da imagem. Tendo em linha de conta que as alterações detectadas em *frames* sucessivas são, normalmente, pequenas, concebeu-se uma solução que divide os raios a enviar para a cena em *frames* sucessivas [Semião06]. Na subsecção 4.1 descreve-se o algoritmo de visão baseado em lançamento de raios e nas subsecções 4.2 e 4.3 apresentam-se soluções para a criação, respectivamente, de pontos alvo e de vectores de deslocamento. Na subsecção 4.4 indicam-se alguns detalhes de implementação.

4.1 Algoritmo de Visão Baseado em Lançamento de Raios

O algoritmo de visão proposto divide o lançamento de raios por *frames* sucessivas. Para este efeito considerou-se uma matriz, M , que em cada linha contém os pontos a utilizar numa *frame*. Ou seja, se, por exemplo, a matriz tiver dimensão $p \times q$, então tem os pontos a utilizar em p *frames*, e em cada *frame* serão lançados q raios na direcção dos q pontos definidos na respectiva linha da matriz. Definiu-se também um vector $A\Delta$ que contém os deslocamentos a aplicar aos pontos de M em sucessivas

iterações. O primeiro deslocamento de $A\Delta$ é $(0, 0)$ de modo a usar sem alterações os pontos da matriz M .

De uma maneira mais formal, consideremos

$M [L, C]$ uma matriz de pontos 2D com L linhas e C colunas, ou seja, relativa a L *frames* com lançamento de C raios em cada *frame*
 e $A\Delta[S]$ um vector com S vectores de deslocamento

A matriz M define $L \cdot C$ pontos alvo a atingir em L *frames*. Combinando os pontos da matriz M com os deslocamentos do vector de deslocamentos $A\Delta$, haverá $L \cdot C \cdot S$ pontos que serão atingidos por raios em $L \cdot S$ *frames*, uma vez que cada vector de $A\Delta$ será aplicado a todos os pontos da matriz M .

Para uma dada *frame* F , o conjunto de pontos atingidos será também usado na *frame* $F+(L \cdot S \cdot K)$, onde K é um valor inteiro positivo. Isto é, ao fim de $L \cdot S$ *frames* volta a repetir-se o mesmo conjunto de pontos alvo.

Assim, o conjunto de pontos a usar na *frame* F é

$$\{M [H, i] + A\Delta[D], i=1,2,\dots,C\}$$

onde

$$H = ((F-1) \text{ mod } L) + 1,$$

$$D = (((F-1) / L) \text{ mod } S) + 1 \text{ e}$$

/ representa a divisão inteira

Consideremos o seguinte exemplo para ilustrar o método adoptado.

Dada a matriz

$$M [2,2] = \{ \{ (0.4, 0.4), (0.6, 0.6) \}, \{ (0.4, 0.6), (0.6, 0.4) \} \}$$

e o vector de deslocamentos

$$A\Delta = \{ (0, 0), (0.01, 0.01), (-0.02, 0.02) \}$$

O conjunto de pontos para onde são lançados raios em cada *frame* é mostrado na tabela 1. Quando o último ponto é atingido, o processo repete-se novamente a partir do primeiro.

Frame id	Conjunto de pontos por frame
1	(0.4, 0.4), (0.6, 0.6)
2	(0.4, 0.6), (0.6, 0.4)
3	(0.41, 0.41), (0.61, 0.61)
4	(0.41, 0.61), (0.61, 0.41)
5	(0.38, 0.42), (0.58, 0.62)
6	(0.38, 0.62), (0.58, 0.42)

Tabela 1

Falta ainda estabelecer a forma de gerar os pontos que constituem a matriz M e os vectores de deslocamento do vector $A\Delta$. Nas subsecções seguintes apresenta-se uma solução possível.

4.2 Geração de Pontos Alvo

O sistema de visão humano capta mais informação na zona central do campo de visão e tem uma visão periférica muito limitada [Peters02]. Para simular a visão humana através de raios lançados do ponto de vista do agente, interessa lançar um maior número de raios na zona central da imagem. Para atingir este objectivo, vai definir-se uma matriz de pontos alvo (MPA) com uma distribuição simétrica nos quatro quadrantes da imagem, considerando distâncias menores entre pontos adjacentes no centro da imagem.

Para definir as coordenadas dos pontos alvo dos raios a lançar a partir do ponto de vista do agente, considera-se um sistema de coordenadas intermédio independente da dimensão do *viewport*. Neste sistema de coordenadas a imagem é um quadrado unitário com origem no extremo superior esquerdo (Fig. 1).

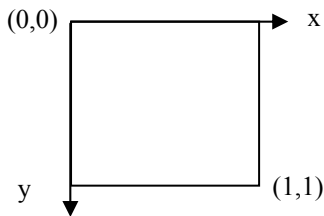


Fig. 1 – Sistema de coordenadas intermédio

Deste modo, para definir a matriz de pontos alvo com coordenadas simétricas relativamente a dois eixos ortogonais com origem no centro do quadrado, basta considerar o mesmo vector, P, para as componentes em xx e em yy, gerando a matriz de pontos como uma grelha rectilinear [Speray90]. Por exemplo, para gerar uma matriz MPA 7x7, consideremos o seguinte vector

$$P=[0.025, 0.25, 0.4, 0.5, 0.6, 0.75, 0.975]$$

obtendo cada ponto da matriz à custa da expressão

$$MPA_{ij} = (P_i, P_j), \quad i, j=1, 2, \dots, 7$$

Assumindo que as coordenadas são simétricas relativamente ao centro da imagem, como acontece no exemplo anterior, basta definir metade das coordenadas do vector P para obter todos os pontos.

Para que a distância entre pontos adjacentes aumente para pontos mais afastados do centro da imagem, vamos considerar que a distância entre os pontos aumenta em progressão geométrica com razão superior a 1. Na fase actual de testes, considerou-se o valor 2 como limite superior da razão da progressão, para garantir que na periferia da imagem também sejam lançados raios.

Como interessa adaptar o número de pontos da matriz de modo a obter um número razoável de FPS, a geração da matriz MPA vai ficar condicionada, não só pela razão, k, da progressão geométrica a aplicar à distância entre os pontos, mas também pelo número de pontos que se pretende gerar. Para gerar um número ímpar de pontos por linha, vamos considerar que o centro da imagem, (0.5, 0.5), pertence à matriz MPA. Como foi referido anteriormente, no vector P há simetria relativamente à

origem. Assim para criar um vector P com um número ímpar de coordenadas, n, uma vez que é conhecida a coordenada central, 0.5, basta gerar $m=(n-1)/2$ coordenadas na metade direita da imagem, ou seja, com valores a variar entre 0.5 e 1. As coordenadas na metade esquerda da imagem são obtidas por simetria relativamente ao ponto central.

Para garantir a geração de m coordenadas em metade da dimensão da imagem, é necessário que a soma das distâncias entre os pontos não ultrapasse 0.5. Além disso, como se pretende aplicar deslocamentos aos pontos da matriz inicial, considera-se uma distância δ entre o último ponto gerado e o lado da janela de modo a que o deslocamento não coloque os pontos fora da imagem. Atendendo que as distâncias formam uma progressão geométrica, determina-se a distância entre dois pontos adjacentes no centro da imagem, d_1 , resolvendo a seguinte inequação

$$d_1*(1-k^m)/(1-k) < 0.5-\delta \quad (1)$$

As distâncias entre coordenadas sucessivas são obtidas pela equação

$$d_{i+1} = d_i*k, \quad i=1, 2, \dots, m \quad (2)$$

Sendo $x_{m+1} = 0.5$ o centro da imagem, as coordenadas à direita do ponto médio são dadas por

$$x_{m+1+j} = x_{m+1} + d_j, \quad j=1, 2, \dots, m \quad (3)$$

e as coordenadas à esquerda do ponto médio são dadas por

$$x_{m+1-j} = x_{m+1} - d_j, \quad j=1, 2, \dots, m \quad (4)$$

No caso do número de coordenadas, n, de P ser par, é necessário criar $m=n/2$ coordenadas em cada metade. Neste caso, o ponto médio da imagem, 0.5, corresponderá ao ponto médio entre os pontos com menor afastamento, d_1 . Ou seja, é necessário garantir que em metade da dimensão da imagem cabe a soma de $d_1/2$ com os termos sucessivos da progressão geométrica das distâncias entre os pontos. Assim, determina-se d_1 resolvendo a seguinte inequação

$$d_1*(1-k^m)/(1-k) - d_1/2 < 0.5-\delta \quad (5)$$

As coordenadas do vector P, correspondentes às coordenadas à direita do ponto médio da imagem, são dadas por

$$x_{m+1} = 0.5 + d_1/2 \quad (6)$$

$$x_{m+1+j} = x_{m+1} + d_{j+1}, \quad j=1, 2, \dots, m-1 \quad (7)$$

e as coordenadas do vector P, correspondentes às coordenadas à esquerda do ponto médio da imagem, são dadas por

$$x_m = 0.5 - d_1/2 \quad (8)$$

$$x_{m-j} = x_{m+1-j} - d_{j+1}, \quad j=1, 2, \dots, m-1 \quad (9)$$

A Fig. 2 mostra o resultado obtido com a geração de uma matriz 7x7 considerando $k=1.5$.

Como relacionar a matriz MPA com a matriz M? Se não houver degradação de desempenho ao tratar numa só *frame* todos os pontos da matriz MPA, pode considerar-se que a matriz M tem apenas uma linha que

contém todos os pontos da matriz MPA. Caso contrário, será necessário repartir os pontos da matriz MPA por várias linhas da matriz M, de modo a serem tratados em diferentes *frames*.

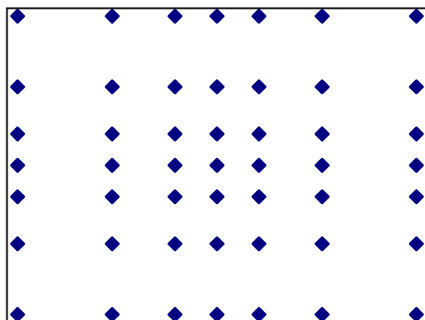


Fig. 2 – Distribuição dos pontos da matriz de pontos alvo, MPA, usando n=7 e k=1.5

4.3 Vector de Deslocamentos

A missão do vector de deslocamentos é provocar a variação dos pontos alvo em iterações sucessivas de modo a atingir o maior número de pontos da imagem.

Como já foi mencionado, a primeira componente deste vector é o vector (0, 0) para permitir em primeiro lugar lançar raios para os pontos calculados.

Os elementos seguintes do vector de deslocamentos devem ser escolhidos de modo a evitar que elementos sucessivos tenham a mesma direcção, para diminuir a probabilidade de o vector de deslocamentos seguir a direcção do movimento do agente em passos sucessivos. Devem também tentar cobrir de uma forma uniforme o espaço circundante aos pontos definidos na matriz MPA.

De modo a preencher estes requisitos foi delineado um algoritmo que tem por base a variação em espiral dos extremos dos vectores de deslocamento.

A geração dos vectores de deslocamento depende de algumas variáveis que podem ser ajustadas para controlar a distribuição dos pontos.

Sejam

$\Delta_{max} = d_1$ – Norma do maior vector de deslocamento, corresponde à menor distância entre quaisquer dois pontos da matriz MPA

N – Número de vectores de deslocamento a calcular

$\Delta = \Delta_{max}/N$ – Incremento da norma entre dois vectores de deslocamento consecutivos

α – Ângulo do vector de deslocamento

α_{Inc} – Incremento ao ângulo do vector de deslocamento

R – Factor para a redução de α_{Inc} ($0 < R < 1$)

Para gerar um novo vector, precisamos de saber a norma do último gerado. Esta informação será guardada na variável Δ_{total} , que é inicializada a zero.

O algoritmo consiste em iterar as seguintes acções N vezes:

Passo 1 - Guardar vector de deslocamento ($\text{Cos}(\alpha) * \Delta_{total}$, $\text{Sin}(\alpha) * \Delta_{total}$).

Passo 2 - Incrementar α em α_{Inc} graus.

Passo 3 - Δ_{total} toma o valor de $\Delta_{total} + \Delta$

Passo 4 - Se α é maior que 360° :

α toma o valor de $(\alpha - 360)$

α_{Inc} toma o valor de $\alpha_{Inc} * R$

Passo 5 - Voltar ao passo 1

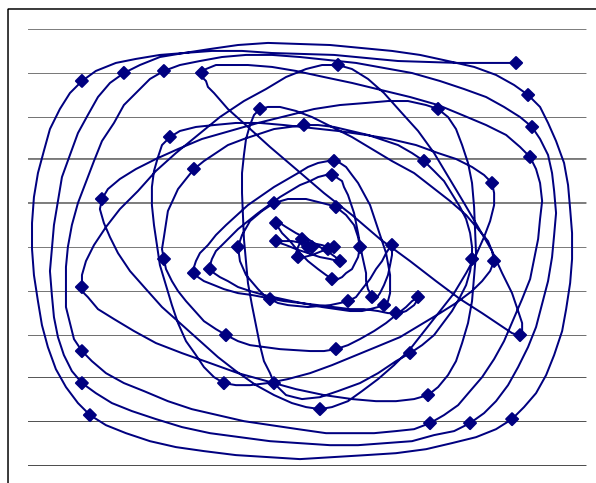


Fig. 3 – Vectores de deslocamento: sequência de extremos finais gerados com R=0.8, N=60, $\alpha = 0^\circ$ e $\alpha_{Inc}=90^\circ$

A Fig. 3 mostra os extremos dos vectores de deslocamento gerados considerando os seguintes valores iniciais: R= 0.8, N=60, $\alpha = 0^\circ$ e $\alpha_{Inc}=90^\circ$. Cada pequeno losango da figura representa o extremo final de um vector de deslocamento cuja origem é o ponto (0,0), que corresponde ao centro da figura. A linha une os pontos que representam os extremos dos vectores de deslocamento pela ordem em que foram gerados.

Como já foi mencionado, a primeira componente do vector de deslocamentos é o vector (0, 0) para permitir que em primeiro lugar sejam lançados raios para os pontos calculados para a matriz MPA.

No passo 4, o ângulo α_{Inc} é actualizado ao fim de cada volta percorrida pelo ângulo α . Deste modo, garante-se que, por um lado, em voltas sucessivas do ângulo α não sejam gerados vectores de deslocamento alinhados sobre semi-rectas com origem nos pontos alvo, e por outro lado, sejam gerados mais vectores em cada volta, à medida que o número de voltas aumenta, para compensar o alargamento da espiral que este gerador tenta aproximar.

Consideraram-se apenas valores positivos para α e α_{Inc} , uma vez que valores negativos destes ângulos apenas modificariam a ordem de geração dos vectores. No entanto, basta alterar, no passo 4, o teste de conclusão de uma volta, para permitir a utilização de valores negativos para qualquer destes ângulos.

4.4 Implementação

Foram realizados alguns testes iniciais para verificar o desempenho do algoritmo, variando o número de objectos na cena e o número de raios lançados para a cena. Como exemplo, mostramos os resultados da tabela 2 com o número de FPS obtidas, fazendo variar o número de objectos e o número de raios. A máquina usada no teste é um Pentium 4 3.40GHz com 1GB de DDR2 RAM e uma placa gráfica NVIDIA GeForce 7800 GTX (Gainward Golden Seal). Foi usado Direct3D9 com uma resolução de 800x600 em *windowed mode*. A cena tem em média 10000 triângulos.

Apesar de os testes terem sido feitos com poucos objectos e lançando poucos raios, o número de FPS é muito elevado, permitindo por isso aumentar, quer o número de raios, quer o número de objectos.

		Número de Raios	
		30	50
Número de Objectos	6	284	264
	12	279	261
	18	267	249

Tabela 2 – resultados em FPS

5. CONCLUSÕES E TRABALHO FUTURO

Apresentou-se um algoritmo de visão sintética baseado em *ray casting*. Este algoritmo divide os raios a lançar por sucessivas *frames*. Deste modo pretende-se otimizar a detecção de objectos sem degradar o desempenho.

Este algoritmo apresenta soluções em termos do cálculo, quer dos pontos a atingir pelos raios, quer dos deslocamentos a aplicar a estes pontos em sucessivas *frames*.

Um dos problemas que normalmente se encontra neste tipo de algoritmos de visão é a não detecção de alguns objectos mais pequenos ou mais distantes. Os algoritmos para geração de pontos alvo e de vectores de deslocamentos têm vários parâmetros ajustáveis. Como trabalho futuro pretende-se realizar testes de modo a adequar estes parâmetros para resolver o problema acima descrito.

Outro aspecto a considerar prende-se com a realização de testes para aprofundar o conhecimento sobre a relação entre o número de FPS com o número de objectos em cena e o número de raios lançados.

O resultado dos testes permitirá conjugar todo o trabalho realizado até ao momento de forma a conseguir um algoritmo adaptável, que consiga manter o número de FPS estável, perto de valores definidos pelo utilizador.

6. REFERÊNCIAS

[Blumberg96] Blumberg, B.: Old Tricks, New Dogs: Ethology And Interactive Creatures. PhD Dissertation, MIT Media Lab, 1996

[Burke01] Burke, R., Isla, D., Downie, M., Ivanov, Y., Blumberg, B.: Creature Smarts: The Art and Architecture of a Virtual Brain, The Game Developers Conference, pp 147-166, 2001

[Carmo05] Carmo, M. B., Cláudio, A. P., Cunha, J. D., Coelho, H., Silvestre, M., Pinto-Albuquerque, M.: Plataforma de Suporte à Geração de Cenas Animadas com Agentes Inteligentes. In Actas do 13º Encontro Português de Computação Gráfica, pp. 79-84, 2005

[hB] <http://www.blender3d.org/About/?sub=Features>

[hJ] <http://jade.tilab.com/>

[hOC] <http://www.codercorner.com/Opcode.htm>

[hOD] <http://ODE.org/>

[hOG] <http://www.ogre3d.org>

[hOGC] http://conglomerate.berlios.de/wiki/index.php/OgreOpcode_Introduction

[Kuffner99] Kuffner, J.J., Latombe, J.-C., Fast Synthetic Vision, Memory and Learning Models for Virtual Humans, Proceedings Computer Animation 1999, IEEE International Conference on Computer Animation, pp 118-127, 1999

[Noser95] Noser, H., Thalmann, D.: Synthetic Vision and Audition for Digital Actors. Proc. Eurographics '95, Maastricht, pp 325-336, 1995

[Peters02] Peters, C., O'Sullivan, C.: Synthetic Vision and Memory for Autonomous Virtual Humans, Computer Graphics Forum, vol. 21, issue 4, pp 743-752, Dez. 2002

[Renault90] Renault, O., Magnenat-Thalmann, N., Thalmann, D.: A Vision-based Approach to Behavioural Animation. Journal of Visualization and Computer Animation, vol. 1, nº 1, pp.18-21, 1990

[Semião06], Semião, P. M., Carmo, M. B., Cláudio, A. P., Implementing Vision in the IViHumans Platform, SIACG 2006, Ibero-American Symposium in Computer Graphics, *short paper*, pp 56-59, 2006

[Speray90] Speray, D., Kennon, S., Volume Probes: Interactive Data Exploration on Arbitrary Grids, Computer Graphics, vol. 24, nº5, Nov. 1990

[Silvestre05] Silvestre, M., Pinto-Albuquerque, M., Carmo, M. B., Cláudio, A. P., Cunha, J. D., Coelho, H.: Platform for the Generation of Virtual Environments Inhabited by Intelligent Virtual Humans. In Proc. ACM ITiCSE'05, *student posters*, pp 402, 2005.

[Terzopoulos96] Terzopoulos, D., Rabie, T., Grzeszczuk, R.: Perception and Learning in Artificial Animals. Artificial Life V: Proc. Fifth International Conference on the Synthesis and Simulation of Living Systems, Nara, Japan, May, pp 313-320, 1996

[Vosinakis05] Vosinakis, S., Panayiotopoulos, T.: A Tool for Constructing 3D Environments with Virtual Agents. Multimedia Tools and Applications, 25, pp 253-279, 2005

Faces – Biblioteca de Expressões Faciais

Janete Faustino Ana Paula Cláudio Maria Beatriz Carmo

Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa
Campo Grande, 1749-016, Lisboa
e25526@alunos.di.fc.ul.pt, {apc,bc}@di.fc.ul.pt

Sumário

A utilização de humanos virtuais inteligentes tem vindo a tornar-se habitual em diversos contextos que combinam contribuições de várias disciplinas, nomeadamente, da Computação Gráfica, da Inteligência Artificial e das Ciências Sociais. A capacidade de transmitir emoções é de fundamental importância para tornar credível a inteligência de um humano virtual.

Neste artigo apresenta-se uma biblioteca gráfica de expressões faciais que define um conjunto básico de expressões e é parte integrante de uma plataforma destinada a gerar ambientes virtuais habitados por humanos virtuais inteligentes.

Palavras-chave

Humanos virtuais, ambientes virtuais, animação comportamental, expressões faciais.

1. INTRODUÇÃO

O desenvolvimento de ambientes de Realidade Virtual ou de Realidade Aumentada é cada vez mais comum, tendo-se tornado um importante assunto de investigação. Estes ambientes são usados em áreas diversas como o entretenimento, a educação, o treino em ambientes industriais, a reconstituição de ambientes históricos, as aplicações médicas e o tratamento de fobias.

A introdução de humanos virtuais inteligentes em ambientes virtuais ou em ambientes reais tem vindo a emergir significativamente, constituindo estes, por si só, um importante tema de investigação onde se combinam contribuições de várias disciplinas, nomeadamente, da Computação Gráfica, da Inteligência Artificial e das Ciências Sociais.

Para que o comportamento e a aparência dos humanos virtuais sejam cada vez mais credíveis, a sua padronização torna-se fundamental, de modo a que a sua reutilização e adaptação sejam possíveis. Para tal, é essencial que no seu desenvolvimento se tenham em atenção as bases científicas, ou seja, o estudo anatómico e psicológico do ser humano sobre os quais assentam a expressividade facial.

Neste documento descreve-se a criação de uma biblioteca de faces de humanóides capazes de expressar emoções. Esta biblioteca, designada por *Faces*, é uma das componentes da plataforma IViHumans, destinada à

geração de ambientes virtuais habitados por humanos virtuais inteligentes [Carmo05]. Pretende-se que esta plataforma possa vir a suportar estudos sobre o comportamento de agentes inteligentes individuais ou integrados em sociedade.

Na secção 2 faz-se o enquadramento do trabalho desenvolvido. Na secção 3 descrevem-se algumas das teorias de cariz psicológico mais marcantes neste domínio. Na secção 4 referem-se os modelos de animação facial mais frequentemente adoptados. Na secção 5 descreve-se o protótipo que foi desenvolvido para visualizar e estender as expressões faciais da biblioteca. A secção 6 apresenta as conclusões e o trabalho futuro.

2. ENQUADRAMENTO

A arquitectura proposta para suportar a geração de cenas animadas com agentes inteligentes na plataforma IViHumans (**I**ntelligent **V**irtual **H**umans), é uma base fundamental no processo de desenvolvimento do projecto *Faces*. Esta é formada por duas camadas: uma camada gráfica e uma camada de Inteligência Artificial (figura 1).

A biblioteca *Faces* será integrada na camada gráfica.

Usando o Blender [httpBlender] como software de modelação, foi desenvolvido um modelo genérico 3D de uma face humana. Sobre este modelo foram feitas variações de modo a gerar um conjunto base de

expressões faciais que correspondem a emoções bem definidas. Foram também geradas as animações relativas à transição entre estas expressões básicas.

Como a plataforma IViHumans usa o OGRE [httpOGRE] como motor de *rendering* tornou-se necessário proceder à exportação do modelo desenvolvido para um formato compatível com o OGRE, exportando as representações gráficas das expressões faciais e as animações entre as várias representações, conforme explicado na secção 5.

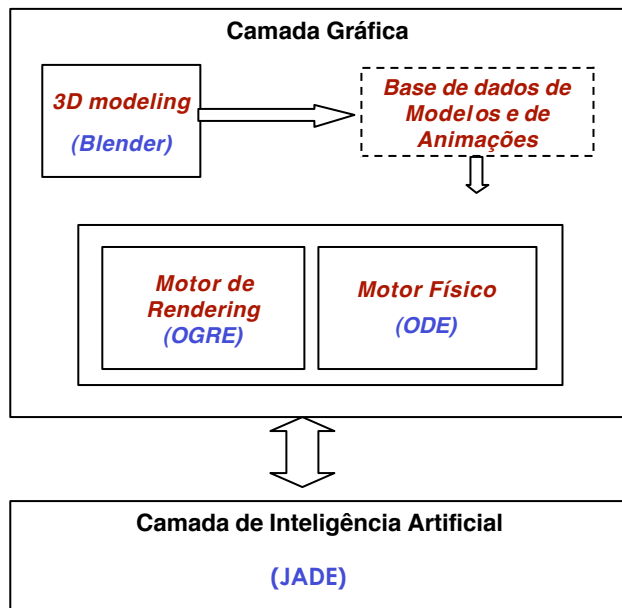


Figura 1: Arquitectura do projecto IViHumans.

No contexto da plataforma IViHumans, a face do humano virtual exprimirá o estado emocional do agente inteligente que lhe corresponde.

3. FUNDAMENTOS

As categorizações emotivas a nível facial são algo que desperta a atenção de psicólogos e cientistas há várias décadas. Segundo Lisetti, Darwin afirmava ser possível fazer uma categorização teórica das emoções, sendo estas consideradas processos mentais e fisiológicos causados pela percepção de eventos de categorias de carácter geral, que geravam um conjunto de sinais internos e externos e diversas combinações de planos de acção [Lisetti02]. Após Darwin esta área de pesquisa começou a despertar cada vez mais interesse, sendo realizados vários estudos neste campo. Um dos psicólogos que mais se destaca é Paul Ekman, afirmando que o ser humano pode assumir seis expressões faciais primárias: alegria, tristeza, raiva, medo, nojo e surpresa [Mandler97]. A cada uma destas expressões faciais primárias correspondem determinados posicionamentos dos músculos e dos elementos faciais (tabela 1).

Apesar de actualmente existirem psicólogos a contrapor a teoria desenvolvida por Paul Ekman, como é o caso de James Russel, esta teoria tem sido a mais adoptada na investigação e desenvolvimento de agentes inteligentes

sociais, bem como na animação de faces virtuais [Mandler97].

Nº	Expressão	Descrição
1	Alegria	As sobrancelhas estão relaxadas. A boca está aberta e os cantos da boca direccionados para cima, na linha de segmento das orelhas.
2	Tristeza	A parte interior das sobrancelhas (perto do nariz) está elevada. Os olhos estão ligeiramente fechados. A boca está relaxada.
3	Raiva	A parte interior das sobrancelhas contrai-se simultaneamente para baixo. Os lábios pressionam-se um contra o outro ou abrem-se ligeiramente, mostrando os dentes.
4	Medo	As sobrancelhas elevam-se simultaneamente e as respectivas partes interiores aproximam-se. Os olhos estão tensos e em alerta.
5	Nojo	As sobrancelhas e pálpebras estão relaxadas. O lábio superior eleva-se ligeiramente de modo assimétrico, fazendo uma pequena curva.
6	Surpresa	As sobrancelhas estão elevadas. As pálpebras superiores estão muito abertas e as inferiores relaxadas. A boca está aberta, descendo significativamente o maxilar inferior.

Tabela 1 – Características de cada uma das seis expressões faciais definidas por Paul Ekman.

4. MODELOS DE ANIMAÇÃO FACIAL

Existem diversos modelos de deformação, ou codificação, facial que têm vindo a ser desenvolvidos ao longo dos anos. Muitos destes modelos são apenas adaptações de modelos generalistas de deformação facial. Alguns baseiam-se meramente na criação de efeitos visuais credíveis sem tomar em consideração as leis estruturais e físicas. Outros são derivados de princípios físicos relacionados com forças, contracções musculares e biomecânica associada à pele, sendo possível categorizá-los de acordo com o mecanismo que controla a geometria do modelo facial. Alguns dos modelos mais conhecidos de deformação facial são a interpolação, os modelos paramétricos, os modelos baseados em acções musculares e o modelo baseado em elementos finitos.

Um dos modelos de deformação facial mais usado baseia-se na norma MPEG-4 [Thalmann04], na qual se especifica:

- o modelo facial em estado neutro;
- um conjunto de pontos de referência – *feature points*;
- um conjunto de parâmetros de animação facial : os *Facial Animation Parameters (FAP)*. A cada um

destes parâmetros corresponde uma acção facial específica que modifica a face no estado neutro. Para definir os *FAP* para modelos arbitrários, o MPEG-4 define unidades de parâmetros de animação facial, os *Facial Animation Parameters Units (FAPU)*, que são aplicáveis a qualquer modelo de face expresso nesta norma. Os *FAPU* são definidos com base em fracções de distâncias entre pontos chaves da face no estado neutro, sendo estas que permitem a interpretação dos *FAP* em qualquer modelo facial de modo coerente.

O MPEG-4 especifica 84 atributos ou pontos de referência para o estado neutro. Estes estão organizados em pequenos grupos como o queixo, os olhos, a boca, entre outros. No total existem 68 parâmetros categorizados em dez grupos relacionados com as várias partes da face. [Thalman04]

5. PROTÓTIPO FACES

A identificação das características faciais que tornam possível ao humanóide expressar emoções e simular a transição entre diferentes estados emocionais foi uma etapa fundamental no início do trabalho. O MPEG-4 foi escolhido para a concretização desta etapa já que se trata de uma norma amplamente divulgada e adoptada.

Usando o Blender como software de modelação e animação 3D foram definidos os 84 pontos de referência do MPEG-4 sobre um modelo facial no estado neutro, procedendo-se de seguida à sua animação de acordo com o modelo de animação facial desta mesma norma. Foram ainda definidas as seis expressões faciais primárias identificadas por Paul Ekman: alegria, tristeza, raiva, medo, nojo e surpresa. Para obter as representações gráficas destas expressões construiu-se uma malha facial para a expressão neutra e foram-se alterando os pontos da malha de modo a obter as características de cada expressão de acordo com a tabela 1 (figura 2). Estas seis expressões faciais primárias são as primeiras a fazer parte da biblioteca de representações gráficas com expressões faciais.

O modelo facial com as respectivas expressões e animações faciais sobre ele definidas foram exportados para o OGRE. Este motor de *rendering* recebe um ficheiro com extensão *mesh.xml* e converte-o em dois ficheiros cujas extensões são *material* e *mesh*. O primeiro ficheiro contém informação relativa à textura do modelo e o segundo contém a biblioteca de representações gráficas com expressões faciais, bem como o modelo genérico da face humana virtual.

Para testar o modelo facial e as animações no contexto do OGRE usou-se uma aplicação disponibilizada na página deste software [httpOGRE]. Esta aplicação, desenvolvida em C++, tem vindo a ser alterada de modo a adicionar um conjunto de funcionalidades que se consideram fundamentais para a plataforma IViHumans.

De entre estas funcionalidades destaca-se a possibilidade de definir novas expressões faciais à custa das já existentes. Segundo o psicólogo Schlosberg, as seis

expressões faciais primárias estão ligadas percentualmente, sendo possível dispô-las num espaço de duas dimensões [Monzani02].

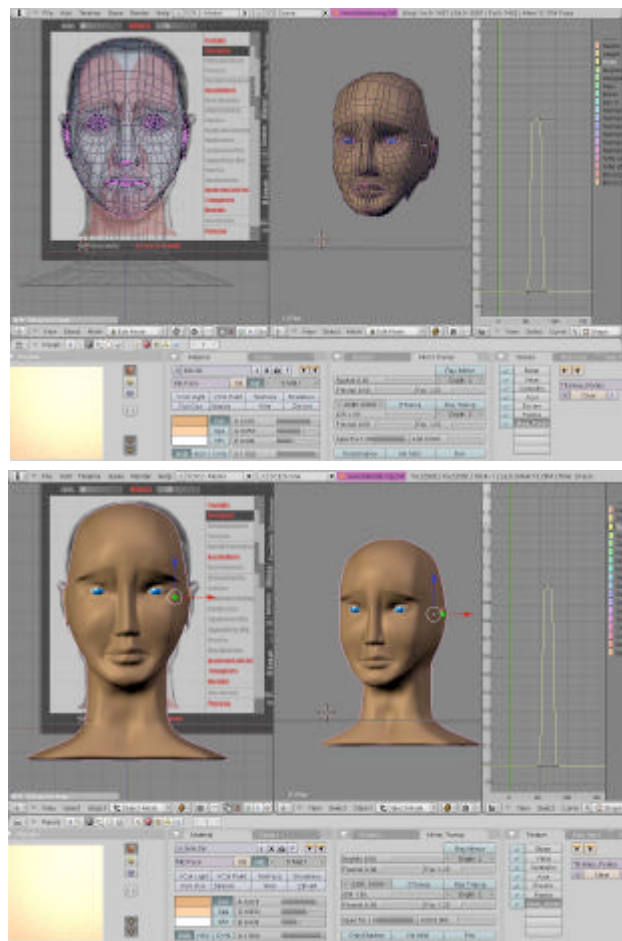


Figura 2: Exemplo da definição de uma expressão facial no modelo desenvolvido – tristeza.

Seguindo esta teoria, Hagen criou uma interface gráfica a qual chamou Disco de Emoções [Monzani02]. Esta possibilitava ao utilizador escolher qual a percentagem que teria cada expressão facial primária na expressão facial final. O perímetro do disco correspondia à percentagem máxima, enquanto o centro correspondia à mínima, ou seja, à expressão neutra (figura 3). [Monzani02]

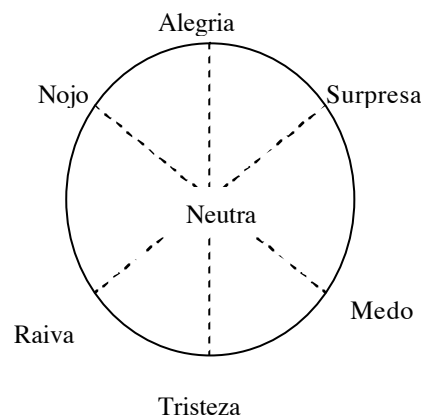


Figura 3: Disco Emocional.

De modo a ir ao encontro desta ideia, a interface da referida aplicação foi adequada de modo a suportar dois modos de actuação: o automático e o manual. Qualquer um dos dois possui as seis expressões básicas faciais e a definição dos pontos de expressão e animação facial segundo a norma MPEG-4. O modo automático permite ao utilizador ver a transição entre os diferentes estados emocionais disponíveis na biblioteca gráfica de faces, no modo manual, o utilizador tem ao seu dispor uma barra de deslocamento para cada expressão facial primária. Cada barra de deslocamento permite controlar a percentagem que cada expressão revela no modelo facial. A aplicação interpola os valores das percentagens escolhidas e gera uma nova expressão facial daí resultante (figura 4).

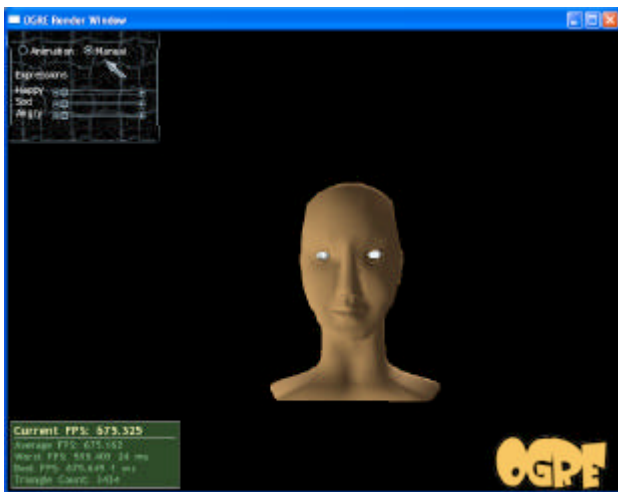


Figura 4: Versão inicial do protótipo da aplicação para testar o modelo facial e as suas animações.

6. CONCLUSÕES E TRABALHO FUTURO

Neste artigo apresentou-se uma biblioteca gráfica de expressões faciais que define um conjunto básico de expressões seguindo modelos e uma norma amplamente aceites pela comunidade científica desta área. A biblioteca é parte integrante de uma plataforma destinada a gerar ambientes virtuais habitados por humanos virtuais inteligentes.

Esta biblioteca é extensível visto que é permitido ao utilizador definir interactivamente novas expressões que poderão, em breve, vir a ser adicionadas à biblioteca.

Serão ainda realizados testes de integração desta biblioteca na plataforma IViHumans procedendo à sua ligação com a camada de Inteligência Artificial.

7. REFERÊNCIAS

[Carmo05] Carmo, B., Cláudio, A.P., Cunha, J.D., Coelho, H., Silvestre, M., Albuquerque, M.P.. *Plataforma de Suporte à Geração de Cenas Animadas com Agentes Inteligentes*, Actas do 13º Encontro Português de Computação Gráfica, pp 79-84, Vila Real, Outubro 2005.

[Faustino05] Faustino, J., *Projecto Faces: Expressão de Emoções em Humanos Virtuais*, Relatório de CEPEI do DI-FCUL, 2006. (A publicar brevemente em <http://www.di.fc.ul.pt/disciplinas/pei/?relatorios>).

[Ghijsen05] Ghijsen, M., Nijholt, A., Heylen, D., Akker, R.. *Facial Affect Displays during Tutoring Sessions*, Workshop at IUI 2005: International Conference on Intelligent User Interfaces, San Diego, CA, USA, 2005.

[httpBlender] <http://www.blender3d.org/>

[httpLabMAG] <http://labmag.di.fc.ul.pt/>

[httpOGRE] <http://www.ogre3d.org/>

[Lisetti02] Lisetti, C. L.. *Personality, Affect and Emotion Taxonomy for Socially Intelligent Agents*, Fifteenth International Florida Artificial Intelligence Research Society Conference, pp 397 - 401, 2002.

[Mandler97] Mandler, G.. *The Psychology of Facial Expression*, Cambridge University Press, 1997.

[Monzani02] Monzani, J.S.. *An Architecture for the Behavioural Animation of Virtual Humans*, Thèse sciences EPFL, Faculté informatique et communications IC, Section d'informatique, 2002.

[Silvestre04] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H.. *Arquitetura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes*, relatório técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-04-7, Julho de 2004.

[Silvestre05] Silvestre, M., Pinto-Albuquerque, M., Carmo, M.B., Cláudio, A.P., Cunha, J.D., Coelho, H.. *Concretização de uma Arquitectura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes*, relatório técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-05-6, Março de 2005.

[Thalmann04] N. Magnenat-Thalmann, D. Thalmann. *Handbook of Virtual Humans*, John Wiley & Sons, 2004.

[When04] Wen, Z., Huang, T. S.. *3D Face Processing: modeling, analysis and synthesis*, Kluwer Academic Publishers, 2004.

Usability in Virtual and Augmented Environments: A qualitative study

Angela Pimentel
Univ. de Aveiro
Aveiro
angelaeika@yahoo.com.br

Paulo Dias
Univ. de Aveiro / IEETA
Aveiro
paulo.dias@det.ua.pt

Frank van Huussen
TU Delft / EWI
The Netherlands
frank@vanhuussen.nl

Pauline van der Horst
TU Delft / EWI
The Netherlands
pauline@ch.tudelft.nl

Jan-Willem Baggerman
TU Delft / EWI
The Netherlands
jwbagger@hetnet.nl

Rafael Bidarra
TU Delft / EWI
The Netherlands
R.Bidarra@ewi.tudelft.nl

Joaquim Madeira
Univ. de Aveiro / IEETA
Aveiro
jmadeira@det.ua.pt

Beatriz Sousa Santos
Univ. de Aveiro / IEETA
Aveiro
bss@det.ua.pt

Abstract

Virtual and Augmented reality are developing rapidly: there is a multitude of environments and experiments in several laboratories using from simple HMD visualization to more complex and expensive 6-wall projection, CAVEs, and other systems. Still there is not yet a clear emerging technology in this area, nor commercial applications based on this technology are used in large scale. In addition to the fact that this is an emerging and relatively recent technology, there is little work to validate the utility and usability of these Virtual and Augmented environments when compared with traditional desktops. However, usability evaluation is crucial in order to design better systems that respond to the users' needs, as well as for finding applications that can really gain from the use of these new technologies.

In this short paper, we present a preliminary qualitative usability study of a Virtual and Augmented Reality environment under development at the University of Aveiro. The main goal is to find clues on how to develop better evaluation methods for systems based on AR/VR technologies.

Key words

Virtual Reality, Augment Reality, Usability, Evaluation.

1. INTRODUCTION

Virtual Reality (VR) is an important emerging technology with applications in many different areas, such as automobile industry, civil engineering, aeronautics, medical science, education and entertainment.

In VR, as for any other product or system, designers should consider usability issues: the ability to carry out tasks effectively, efficiently and with satisfaction by their intended users [Marsh99]. That is, design systems as simple and usable as possible. Although VR technology has improved in the last few years, it is still necessary to define methods and parameters to test and evaluate the usability of those systems during their development and implementation cycle [Paelke00]. If usability evaluation tools, methods, and techniques are well defined for 2D environments, there are relatively few for the evaluation of 3D Virtual and Augmented Reality systems [Marsh99].

It is a common idea that technology and Virtual Reality systems can facilitate the execution of tasks. But, if they

are not well designed, users will face usability problems, such as motion sickness, difficulty of navigation or spatial disorientation [Ware00][Smith04]. Furthermore, there are other problems specific to these technologies: high cost, performance and availability of software, database management, input and output devices, data visualization, mathematical description and representation of physical processes. Due to all these limitations, many experts still find traditional desktop systems easier to use than virtual systems [Demiralp06]. Usability evaluation also faces another problem: few companies are interested in investing money and resources to evaluate usability; this may result in systems that are not adapted to intensive work and are used only due to innovation or curiosity without providing any real benefit.

The work described here is a first step towards evaluating the usability of a low cost Virtual and Augmented Reality environment under development at the University of Aveiro [Moreira05]. This is a generic prototype for exploration of three-dimensional models and data that uses

a head-mounted display (HMD), a video camera, orientation sensors and an accelerometer.

2. TEST ENVIRONMENTS

Our environment can be used in two different modes: Augmented and Virtual Reality. In what follows we describe the experiments and applications developed to test both modes. Results are then presented in section 3 for the Augmented Reality environment. The Virtual Reality experiment we describe is still under development.

2.1 Hardware

The hardware consists of a HMD *i-glasses SVGA Pro* with resolution of 800x600 pixels, an orientation sensor (tracker) *InterTrax 2* from *InterSense* with three degrees of freedom (yaw, pitch and roll) (Figure 1).

For the Augmented Reality environment, the Virtual Reality hardware was adapted by simply adding a camera to the system. The use of a HMD with a micro camera mounted on it prefigures video see-through Augmented Reality applications.

The camera used is a wireless analog micro-camera with a CMOS sensor of approximately 330 lines, working at 1.2GHz. Given the small dimensions and weight of the camera, it was possible to mount it directly at eye-level on the HMD and power it from a 9V battery.



Figure 1: HMD with tracker and micro-camera

2.2 Augmented Reality

Augmented Reality (AR) environments combine real scenes with scenes generated by the computer, making possible the visualization of data, models and meta-information [Pan05]. The AR setting used in this work is the most common one: Video See-Through, which uses a conventional HMD to render Virtual elements on the images captured by a camera.

Our experiment consists in showing to different users data from their own field of expertise within an Augmented Reality environment. Two different configurations are used: in the first one, called AR (Augmented Reality), the user sees, through the HMD, virtual models or data aligned with a real marker (a black square with known geometry) captured by the camera. In the second one, called AD (Augmented Desktop), the same augmented models or data are shown in a desktop display and the view is updated according to a hand-held video camera the user can move around the markers.

Our Augmented Reality prototype uses the ARToolkit [Billinghurst99]: a graphical toolkit capable of detecting the orientation of markers in a scene and render virtual objects according to the orientation and position of the

markers. The orientation of the camera is also extracted directly from the analysis of the known markers. For the processing and representation of different types of 3D data available, since ARToolkit only provides rendering of VRML models, we use The Visualization Toolkit (VTK) [Schroeder98]. This toolkit is useful since it offers hundreds of classes for the manipulation and visualization of 3D data, import of models and interaction with devices. Basically, we used ARToolkit only to extract the camera pose from the images and render the object correctly according to the marker in VTK. This configuration provides much more flexibility and allows testing several different datasets and visualization methods with minimal effort.

2.3 Virtual Reality

The VR test environment is an immersive game, which consists of corridors defining a maze in which the player has to navigate. In order to complete the game, the users have to get 21 objects in a pre-defined period of time, which is 5 minutes. The Virtual Reality environment (VR) uses a HMD for display, mouse buttons and orientation sensors for interaction. The environment that is going to be compared with the virtual environment is a traditional desktop environment (VD), where the user plays the same game using mouse and keyboard as interaction devices and the desktop display.

Regarding the VR software, most of the programming was done in C++ using OGRE [OGRE], a well-known open-source rendering engine. Some additional code was written to perform collision detection and counting/recording of events within the game. The user interface is similar in both Desktop and Virtual Environments. The mouse/tracker is used for the orientation of the head (where the player is looking at), whereas the mouse buttons/arrows of the keyboard are used for the movements inside the maze (where the user goes). This experiment is still under development.

3. USABILITY STUDY

3.1 Experiment Description

The first experiment consists in the visualization of data from a mechanical model of water flow around a ship hull. The pressure at each point is mapped through colour, and arrows are used to represent the velocity of the water around the ship hull (see Figure 2).

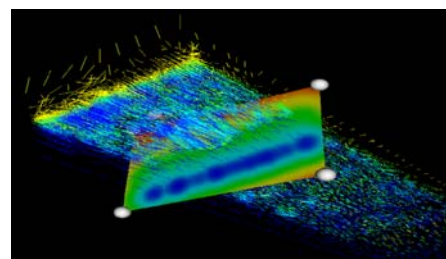


Figure 2: The water-flow data set.

We use four different representations of the same data to provide the users with several viewpoints and scaling factors to make visualization easier:

- The complete data viewed from above;
- A scaled version of a sub-set viewed from the side (corresponding only to the main area of interest);
- The same sub-set viewed from above;
- The sub-set without any marker and camera image, having a totally black background.

Those four different representations were visualized in the Augmented Reality (AR) and in the Augmented Desktop (AD), in order to compare the differences between these two environments and assess which one provides better manipulation, orientation and/or image visualization for the users.

The eight stages of the experiment are presented in Table 1.

1.	AR with complete data viewed from above
2.	AR with sub-set data viewed from the side
3.	AR with sub-set data viewed from above
4.	AR without markers (only data was represented)
5.	AD with complete data viewed from above
6.	AD with sub-set data viewed from the side
7.	AD with sub-set data viewed from above
8.	AD without markers

Table 1: Different conditions of the experiment

The experiment started with the AR environment and then proceeded to AD, since we expected users would prefer the AD due to the better image resolution and contrast of the desktop display.

The AR experiment is presented in Figure 3. The user sees the model lying on the marker and can interact easily by moving the head as well as the marker to change the viewpoint. Figure 4 is a screenshot of the monitor in AD mode. In this experiment, the user holds a camera in his hand and moves it around the marker to change the viewpoint while observing the updated image in the monitor.

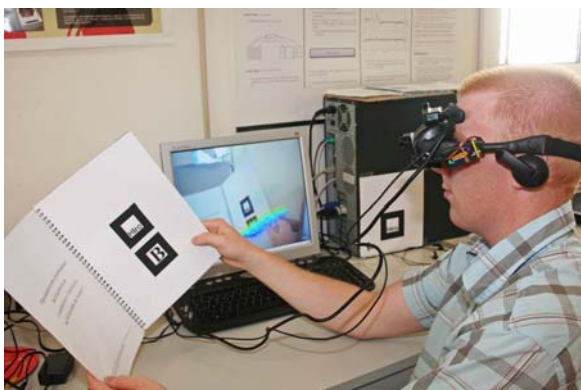


Figure 3: The AR experiment with a clone copy of the image seen by the user on the monitor.

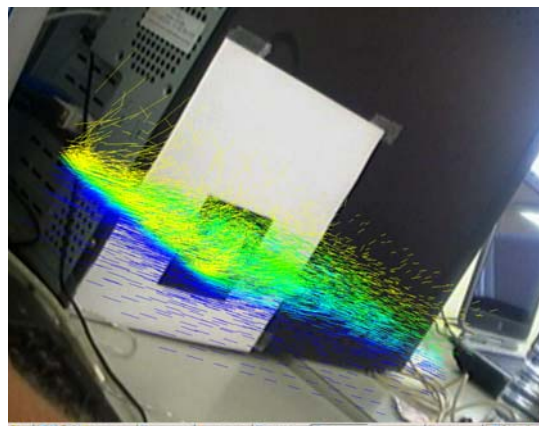


Figure 4: The AD experiment: image captured by the hand-held camera and projected on the monitor.

3.2 Evaluation Methods

In this experiment, we used a qualitative evaluation: users were asked to use the application and then express their opinions by talking to the evaluator while or after they are performing the tasks. The evaluator can comment and counterargument, but at the end users state their overall preferences.

3.3 Participants

Two application domain users were invited to test the eight situations listed above. One is a Professor at the Department of Mechanical Engineering at the University of Aveiro, and the other is a doctoral student. They are both male and it was their first contact with a virtual environment.

3.4 Results

First of all, we were surprised with the positive response of the participants perhaps this was due to the enthusiasm of using a new and attractive form of visualization. Users showed great interest in the application, suggested some improvements and proposed future work with other kinds of data sets.

When the experiment was designed, we expected the users could get lost on the fourth and eighth experimental condition (with no background), since there are no markers to guide them, which was not the case since users found that manipulation of the marker was intuitive enough.

Concerning manipulation and orientation issues, both users concluded that the Augmented Reality environment (AR) with HMD was more intuitive than the Augmented Desktop (AD).

On the other hand, AD was found better than AR regarding the data visualization, due to better resolution and contrast, a result that we already expected.

To conclude the experiment we asked the user suggestions on how to improve the system. The more relevant are:

1. The possibility to navigate inside the model in a totally immersive Virtual environment;
2. A pointer to indicate specific data features to other users;
3. An option/button that would allow the user to zoom in or out the data;
4. Although they did not get lost on the fourth and eighth experiments, they suggested displaying a thin square representing the border of the marker to help its localization without disturbing the viewing of the data.

Their overall conclusions were that for short periods of time, as when somebody wants to show something to a user that is not used to this kind of data/systems, the Augmented Reality (AR) makes data manipulation easier since the HMD interaction is more intuitive. However, for long periods of work the Augmented Desktop (AD) environment could be better than the Augmented Reality (AR), because the visualization is more precise and less tiring.

As a consequence of users suggestions, two additional experimental conditions were implemented. One adds a virtual thin white line around the marker in the fourth and eighth experimental condition. The other shows the camera image instead of the black background, when the user loses sight of the marker. Finally we also added stereoscopic vision.

These modifications were presented to the same users. The results were that the thin square does help to manipulate the object without losing sight of the marker, but when it disappears from the camera field of view, seeing the real image helps to find the marker faster. So the users preferred to combine the two suggested situations. Regarding stereoscopic vision, although the users think it can add interesting information, it does not produce a significant improvement in this particular data set (probably because it was a vector dataset). The setting of the distance between eyes was relevant since one of the users had great difficulties to sense depth until this distance was set to a smaller value [Ware00].

4. CONCLUSIONS AND FUTURE WORK

The technology of Virtual Reality is still a challenge to outdo. The image contrast and resolution are still much better in a desktop display, which is also less tiring compared with the HMD. The main advantage of the Augmented Reality in our experiment was the natural and easy interaction when compared with the hand held camera: the user loses sight of the marker more often in this latter configuration.

Even though this work is still in an early stage, some additional experiments involving brain and lung data and

domain expert users have been conducted with similar results to the ones described before.

In the VR case, some tests comparing the virtual and desktop environment have been performed. The evaluation methods used were a combination of qualitative and quantitative studies. On the qualitative study, as in the Augmented Reality experiments, the evaluator asked the user's opinion, while the quantitative study measured variables, such as time, number of collisions and falls, speed, distance covered, and objects caught.

5. ACKNOWLEDGEMENTS

The work of Frank van Huussen, Pauline van der Horst and Jan-Willem Baggerman was made possible thanks to an Erasmus Program exchange between UA and TUDelft. We would like to thank Filipe Teixeira Dias and Alexandre Cruz for their time and contribution to this study.

6. REFERENCES

- [Billinghurst99] Billinghurst M, Kato H. Collaborative Mixed Reality. In *Proceedings of International Symposium on Mixed Reality (ISMR '99)*. (1999), pp. 261-284.
- [Demiralp06] Demiralp C, et. al. CAVE and Fishtank Virtual-Reality Displays: A Qualitative and Quantitative Comparison, *IEEE Transaction on Visualization and Computer Graphics*, Vol. 12 No.3, (May/June 2006), pp. 323-330.
- [Marsh99] Marsh T. Evaluation of Virtual Reality Systems for Usability. *Doctoral Consortium CHI*, Oxford Univertisy Press, (1999), pp. 15-20.
- [Moreira05] Moreira P. et al. Um ambiente de Baixo Custo para Aplicações de Realidade Virtual e Aumentada. *Proceeding of EPCG 2004. Universidade de Trás os Montes e Alto Douro*, Portugal, 12-14 October 2005. pp.145-149.
- [OGRE] OGRE 3D, April 2006. <http://www.ogre3d.org/>
- [Paelke00] Paelke V. Agents as Building-Blocks for Usability Tests in Multi-User 3D Environments. *CHI Interactive Posters*, (April 2000), pp. 173-174
- [Pan05] Pan Z. et al. *Virtual reality and mixed reality for virtual learning environments*. Elsevier Ltd, (2005), pp 20-27.
- [Schroeder98] Schroeder W. et al. *The Visualization Toolkit- An Object Oriented Approach to 3D Graphics*. 2nd ed., Prentice Hall, (1998).
- [Smith04] Smith P. S., Marsh T., *Evaluation design guidelines for reducing user disorientation in a desktop virtual environment*, Spring-Verlag London Limited, (2004).
- [Ware00] Ware C. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.

Information Visualization Attributes and Operators for a Web Log Based Management System

Florin Zamfir
Univ. de Aveiro
Campus de Santiago, Aveiro
florin@ca.ua.pt

Óscar Mealha
Univ. de Aveiro
Campus de Santiago, Aveiro
oem@ca.ua.pt

Beatriz Sousa Santos
IEETA / Univ. de Aveiro
Campus de Santiago, Aveiro
bss@ieeta.pt

José Nunes
Univ. de Aveiro
Campus de Santiago, Aveiro
jnunes@ca.ua.pt

abstract

This paper presents the theoretical basis of a web log based management system used for research purposes. The main focus is related with information attributes and operator definitions used for information visualization analysis and manipulation. Web sites and portals are becoming complex organic systems difficult to monitor usage patterns and control navigational and structural problems. In many day to day routines, severe inefficient and ineffective situations are ignored or understood a little too late. The system mentioned in this paper represents a research framework with the goal of solving information and communication management problems in a web site/portal, using an information visualization approach, with direct manipulation adjustment and problem identification mechanisms.

Although the specific contribution of this paper is theoretical, some system usage screenshots were added to clarify the purpose of some attributes, definitions and operators.

keywords

information visualization, information and communication management, web logs.

1. INTRODUCTION

Starting from the following premise presented by [Ivory2002] “Using quantitative measures of the informational, navigational, and graphical aspects of a Web site, a quality checker aims to help nonprofessional designers improve their sites.” and using a complementary approach, our goal is to analyze existing hypermedia structures and provide legible visual feedback for their usage patterns. Usability issues / clues are discovered using quantitative measures of informational, navigational and graphical aspects of a website, enriched with a subset of qualitative based measures. Qualitative measures are derived from usability guidelines, combined with metrics derived from empirical analysis of web sites involving web site dynamics, navigational paradigms, usage scenarios, contents classification, etc. In order to achieve this, information attributes must be defined, information visualization operators conceptualized and system components developed into a unified framework with tightly coupled inspection methods.

For the purpose of this paper, that presents the theoretical basis of a web log based management system used for

research purposes, it is important to mention that information visualization is a complex process that transforms raw data into visual features. It combines several techniques to collect, filter and represent the information. According to [Healey2001], scientific visualization is the conversion of collections of strings and numbers into images that allow viewers to perform visual exploration and analysis. The information has to be collected and stored in accessible formats for interrogations. During this process, filters are applied to validate the contents. Afterwards, the information is available for exploration and inspection in a natural form (raw data) or organized in logical relational structures. Depending of the amount of information to be represented, several techniques are available to transform raw data into visual features.

One important contribution of this work is related with the approach to combine quantitative and qualitative measures for website structure, contents and usage into a integrated automated analysis tool, meant to help decision makers inside institutions reorganize their institutional websites according to the feedback provided by such a tool. The identification and association of several

attribute sets to page, text, link and graphical elements, as detailed in section 3 – Theoretical Information Framework, and the association of a visual transformation function to a unique structural attribute is somehow innovative and simplifies the representation of information.

We use visual features to provide help in answering the following general question: How is the site used? Starting from this general question, we can detail a specific sub-set of questions, whose answers might provide useful feedback: Who is using the site? What are the site areas / sectors / pages of interest? What statistical information can be obtained from the log files? What navigational behaviors can be detected by analyzing the associated usage statistics?, etc.

Exploring the enormous capacities of the human visual system, we present the information through different

visualization methods, to provide help for answering the following concrete questions: Which are the areas with problems? What usability problems can be identified and at what level (content relations, semantic, navigational, design layouts, etc.)? How does the visual organization of the site influence user’s navigational decisions?

The following sections describe the conceptual model of a proposed web log based data collection and visualization system and its theoretical information framework that includes; visualization metrics and taxonomy, data representation and raw mapping functions to visual features. System usage screenshots are used as examples to complement theoretical definitions and representations.

2. CONCEPTUAL SYSTEM MODEL

A conceptual model for the visualization subsystem, as introduced by [Mealha2004], is represented in Figure 1.

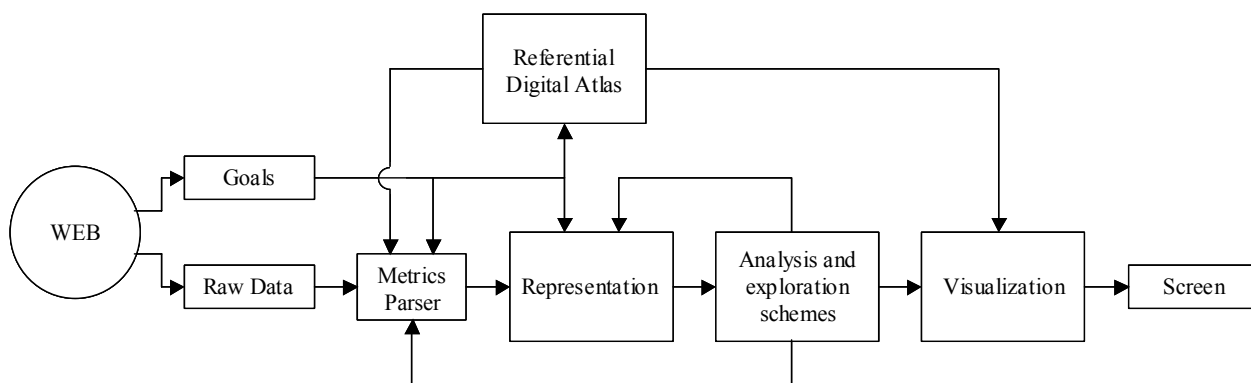


Figure 1 Conceptual System Model

2.1 Description of modules

1. *Web*: represents the website with all its available services, pages and organization;
2. *Goals*: represents the set of specific results to be obtained and all the intermediary steps required for each of the goals to be attained;
3. *Raw Data*: represents the basic information available on its primary state. The information is available from several sources as: website contents, usage log files, eye tracking, motion tracking or interception systems, etc. This involves all the information as stored on its primary source, without any preliminary processing or filtering applied;
4. *Metrics Parser*: this module is designed to filter and extract valid information from the entire store of raw data. One example of information filter and extraction is the detection of usage session based on a specific set of attributes that uniquely identify a user session as a time window within the active time of a specific client, or “A delimited set of user clicks across one or more Web servers.” [WCTD2006].

Considering: (1) $\Delta T_k = \sum_{i=1}^n \Delta T_{j,i}$, the sum of all n

time windows of a specific client j that accessed the web server, $\Delta t_k \in \Delta T_{j,i}$ = the time periods between subsequent requests $t_{k,i}$ of a specific client’s time window i ; a request $t_{k,i}$ is considered as part of the same session i if (2) $\Delta t_{k,i+1} - \Delta t_{k,i} \leq \delta$, where δ = the maximum accepted time window for two subsequent requests to be considered in the same session. In this case, $\Delta T_{j,i}$ is considered a user session.

A 30 minutes δ represents the most common value for the definition of a time window (session) for most of the implementations: according to IFABC Global Web Standards [IFABC2006] a *Session* (or a *Visit*) is a “series of one or more Page Impressions, served to one User, which ends when there is a gap of 30 minutes or more between successive Page Impressions for that User.”

5. *Referential Digital Atlas*: represents the information library that contains all relevant information for the system. It is represented using a specific format that supports the identification of all $A_i, A_{T,j}, A_{L,k}, A_{G,m}$ elements presented in the next section – Theoretical Information Visualization framework;

6. *Representation*: this module includes all available methods for representation of information used to produce visualizations. Several visualization methods are available (2D, 3D, etc.), each design within a specific set of goals;
7. *Analysis and exploration schemes*: the representations obtained from the previous module are combined with additional attributes to support direct manipulation of specific parameters that have a direct impact on the results produced by the visualization. This module provides interaction means with the visualization methods, allowing customizable representation to be produced;
8. *Visualization*: this module combines the results of all previous modules with the aim to display a visual representation of the processed information, which provides adequate interaction and inspection mechanisms;
9. *Screen*: represents the media used to display the information and can be implemented using traditional screens or sophisticated representation technique as virtual 2D/3D glasses, etc.

A technical transpose of the conceptual model is presented in Figure 2. As presented in [Zamfir2004] and [Mealha2004], the system model is composed of five active components, one database and a parameterization module used to filter the information. Three of the active modules (analyzer, compiler and interceptor) are used to gather/intercept the information stored / vehiculated on / to the web server, while the fifth component, the visualizer, maps the functionality of the visualization component of the conceptual model.

Modules *Web*, *Goals*, *Raw Data* and *Metrics Parser* have direct mapping to *web server*, *analyzer*, *interceptor* and *compiler* components, module *Referential Digital Atlas* is represented by the *database*, while modules *Representation*, *Analysis and exploration schemes*, *Visualization* and *Screen* are represented by the *visualizer* component.

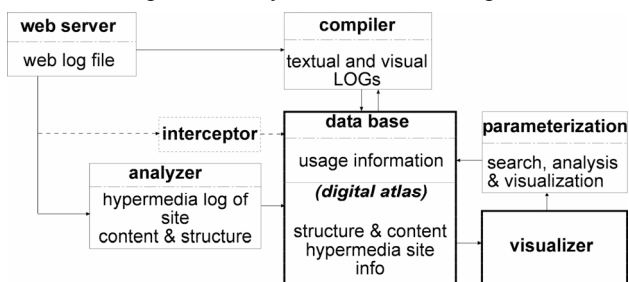


Figure 2 System Model – Conceptual Components

3. THEORETICAL INFORMATION FRAMEWORK

Our goal here is to represent the website contents as it is provided to its final users, in a web browser. For this purpose, we considered the structure of a website as a union of two sets: pages and interconnections, usually called hyperlinks: (3) $W = (P, H)$.

P is the set of pages: (4) $P = \{P_1, \dots, P_n\}$ where n = number of site pages; H is the set of all possible interconnections between all website pages.

H is the set of hyperlinks: (5) $H = \{H_1, \dots, H_m\}$, where m = number of website interconnections. A hyperlink H_i is defined as an edge in the graph W that represents the website: $H_i = (p, q) \in P$, where p represents the origin page (also called *referrer* page) and q represents the child page (also called *referenced* page).

3.1 INFORMATION ATTRIBUTES

One potential approach inspired by Healey's [Healey2001] was introduced in [Mealha2004] to associate attributes to each hyperlink element present on a webpage: each page $P_t = \{h_{t,1}, \dots, h_{t,n}\}$, contains n hotspots; each hot-spot, $h_{t,n}$, of any specific page has at least $m > 1$ attributes, $A = \{A_1, \dots, A_m\}$. Using Ivory's [Ivory2002] approach, a page is composed of *Text elements*, *Link elements*, *Graphic elements* and a page P_i can be specified as the union of the three sets:

(6) $P_i = T_{i,p} \cup L_{i,q} \cup G_{i,r}$, where $L_{i,q}$ represents the set of hotspots. According to Faraday's [Faraday2000] classification, several page elements capture user's attention during the first visual scan of a page, some attributes like color, size, position, text style, etc., being able to directly influence user's navigational decision.

For the purpose of this works, an approach similar to (6) was adopted, even if all three presented solutions are conceptually similar. In conformance with previous definitions, the attributes definition can be extended to all elements that correspond to a web page: considering the page P_i with content elements $T_{i,p} \cup L_{i,q} \cup G_{i,r}$, a set of attributes can be associated to each of these

$$(7) A_T = \{A_{T,1}, \dots, A_{T,w}\},$$

$$(8) A_L = \{A_{L,1}, \dots, A_{L,v}\},$$

$$(9) A_G = \{A_{G,1}, \dots, A_{G,w}\}.$$

A_T = the set of attributes associated to Text elements, A_L = the set of attributes associated to Link elements and A_G = the set of attributes associated to Graphical elements; u, v, w = number of elements associated to Text, Link or Graphical elements.

Seven structural attributes were identified for the web page itself and another three sets, each of five structural attributes, for each of text, link and graphical elements, as follows:

- A_I = URI (*Unique Resource Identifier*) of the page. For dynamic or personalized pages, a unique identifier can be generated to cope with the uniqueness criteria;

- A_2 = dimensions of the page in a fixed size browser window (e.g. a possible reference resolution can be considered 1024x768 browser window sizes);
- A_3 = minimum *site level* the page belongs to;
- A_4 = page raw content size in bytes;
- A_5 = page raw stored copy;
- A_6 = snapshot image of the page that is referenced as it is presented to the user;
- A_7 = dynamism type (static, dynamic, personalized, etc.).

For Text elements, a set of five structural attributes for the A_T set were identified:

- $A_{T,1}$ = URI (*Unique Resource Identifier*) of the page it belongs to;
- $A_{T,2}$ = font characteristics (size, style, spacing, color, paragraph, reading order);
- $A_{T,3}$ = position, type and size of the area occupied by the text in the page;
- $A_{T,4}$ = culture (language);
- $A_{T,5}$ = dynamism type (static, dynamic, personalized, etc.).

For Link elements, a set of five structural attributes for the A_L set were identified:

- $A_{L,1}$ = URI (*Unique Resource Identifier*) of the page the hotspot (link) belongs to;
- $A_{L,2}$ = position and size of the area occupied by the hotspot in the page;
- $A_{L,3}$ = type of hotspot (textual link, image link, composed graphical link, etc.);
- $A_{L,4}$ = URI of the page that is referenced;
- $A_{L,5}$ = dynamism type (static, dynamic, personalized, etc.).

For Graphical elements, a set of five structural attributes for the A_G set were identified:

- $A_{G,1}$ = URI (*Unique Resource Identifier*) of the page it belongs to;
- $A_{G,2}$ = occupied area type, position, size, colors;
- $A_{G,3}$ = motion / movie presence;
- $A_{G,4}$ = interactive or not;
- $A_{G,5}$ = dynamism type (static, dynamic, personalized, etc.);

Basically, we used Healey's [Healey2001] definitions to achieve visualization using a function $M(V, \phi)$ that transforms raw-information into a visualization feature V_j on the visual workspace: $V = \{V_1, \dots, V_r\}$ represents the visual features that are used to display the attribute(s)

(10) $A_j.\phi : A_j \rightarrow V_j$, in other words, maps the A_j domain into the viewable feature domain V_j ; A_i domain is the domain of attributes specific for a web page, $A_{T,j}$, $A_{L,j}$, $A_{G,j}$ are the domains of attributes for text, link and graphical elements of a page j .

(11) $[M_1(V_1, \phi_1) \ M_2(V_2, \phi_2) \ \dots \ M_f(V_f, \phi_f)]$ represents a vector of functions that transform raw information into visual features and in our particular case f depends of the number of visualization methods that can be achieved based on each of the attribute sets to be visually represented.

According to Farraday [Faraday2000], the visual analysis of a webpage is achieved in two phases: search and scanning. These phases can be analyzed based on the visual attributes that uniquely identify each page element. Subsets of the translation functions domains (10) and (11) can be used to code these visual attributes and visually highlight content classification based on focusing and attention capture information.

Unlike Farraday's approach that is based on guidelines derived from the use of eye-tracking techniques that identify which interface elements attract user attention, for our case of study, the information can be achieved by analyzing live experiments data produced by an eye / motion tracking or interception system, as exemplified by Card [Card2001].

To code the focusing and attention levels of page elements during search and scanning phases, a subset of attributes A_F can be defined as a subset of:

$$(12) A_F \in A_T \cup A_L \cup A_G.$$

The structural attributes sets can be extended accordingly to usage information that can be extracted from the log files and / or controlled experiments, [Drott1998], [Zamfir2004], [Fraternali2003], or usage information collected using visual / motion tracking or interception systems. The aim here is to be able to visually represent website usage information, navigational patterns, interface design layouts or other types of raw information.

For the webpage and its contents, structural attributes are extended with statistical usage information to support statistically rich information transformations of statistical raw data into visual features:

For the webpage, the following extended attributes were identified:

- A_8 = number of times this webpage was visited during a specific time period;
- A_9 = frequency of requests for the webpage in the given time interval;
- A_{10} = minimum amount of time the page was analyzed by the clients, using the $\Delta t_{k_{\min}} \in \Delta T_{j,i}$ of two subsequent request;
- A_{11} = maximum amount of time the page was analyzed by the clients, using the $\Delta t_{k_{\max}} \in \Delta T_{j,i}$ of two subsequent request;
- A_{12} = most analyzed areas of the page (potential information generated using an eye / motion tracking or interception system);

For Text elements, a sixth and seventh attributes were identified for the A_T set:

- $A_{T,6}$ = the average time users spent to analyze the text element: $\Delta T_{kta} \in \Delta T_{j,i}$;
- $A_{T,7} \in A_F$ = classification of the text element on a scale, regarding focusing and attention capture.

For Link elements, four other attributes were identified for the A_L set:

- $A_{L,6}$ = the average time users spent to analyze the link element: $\Delta T_{kla} \in \Delta T_{j,i}$;
- $A_{L,7}$ = average time to go from one page to another using this hotspot: $\Delta T_{klj} \in \Delta T_{j,i}$;
- $A_{L,8}$ = number of times this link element was clicked (visited) during a specific time period;
- $A_{L,9} \in A_F$ = classification of the link element on a scale, regarding focusing and attention capture.

For Graphical elements, two other attributes were identified for the A_G set:

- $A_{G,6}$ = the average time users spent to analyze the link element: $\Delta T_{kga} \in \Delta T_{j,i}$;
- $A_{G,7} \in A_F$ = classification of the link element on a scale, regarding focusing and attention capture.

Note that some of these extended attributes are complex entities that are represented using subsets of elementary attributes: (13). Elementary attributes have direct mapping to relational structures as relational databases or similar entities. Relational or Object Oriented Databases (RDB or OODB) are ones of the most adequate systems for this purpose.

$$(13) A_z = \sum_{i=1}^n a_{w,i} | a_{w,i} \rightarrow \Delta_{x,i}, \text{ where } A_z \text{ is the complex attribute } z, a_{w,i} \text{ are elementary sub-attributes, and } \Delta_{x,i} \text{ represents the domain of atomic data types that can be directly represented in a relational data system.}$$

Each visualization method makes use of a limited subset of structural and extended attributes to represent its information. Visual inspection methods may use a subset or complete set of visual elements defined by $M_f(V_f, \phi)$ to map raw information into visual features at a time. Complementary information can be represented at the same time, to highlight specific goals.

Based on 0 and 0, a potential representation for the web structure might consider some of the attributes sets previously described. Four visual elements $V_1 = \{V_{1,1}, V_{1,2}, V_{1,3}, V_{1,4}, V_{1,5}\}$ can be defined to represent the website structure and usage information as follows:

Based on 0 and 0, a potential representation for the web structure might consider some of the attributes sets previously described. Four visual elements $V_1 = \{V_{1,1}, V_{1,2}, V_{1,3}, V_{1,4}, V_{1,5}\}$ can be defined to represent the website structure and usage information as follows:

- $V_{1,1}$ = representation of a page as a tree node with image thumbnail or symbol;
 - $\phi_{1,1}$: represents the page as a tree node with optional page image thumbnail obtained from A_6 attribute information or as / (or on) a basic 2D/3D geometric symbol (color circle or square);
 - $V_{1,2}$ = connecting line between pages or nodes;
 - $\phi_{1,2}$: uses $A_{L,1}$ and $A_{L,4}$ attribute information to represent and connect nodes as directional lines, uses $A_{L,8}$ to calculate line thickness and statistical connectivity;
 - $V_{1,3}$ = visits info (statistical page visits, analysis times, etc.);
 - $\phi_{1,3}$: visits information is visually embedded into the page representation using color, thickness or shape coding techniques, or it can be driven by a roll-over event over the page object, visual feature that pops up a window with statistical information. Several extended attributes can be used to highlight visits information: A_8, A_9, A_{10}, A_{11} ;
 - $V_{1,4}$ = connectivity info (statistical, URLs, etc);
 - $\phi_{1,4}$: driven by a roll-over event on the connecting line, this visual feature feedbacks a pop up window with statistical information, URLs, etc. related to the nodes or to their inter-connectivity. Different graphical representations are user: textual, percentage, progress bars, etc. Several attributes can be used to highlight usage information: $A_8, A_9, A_{10}, A_{11}, A_{L,6}$, etc.;
 - $V_{1,5}$ = an image thumbnail;
 - $\phi_{1,5}$: driven by a roll-over event, it draws a thumbnail with the contents of the page represented by the network or tree node with a semi-transparent factor. It is always visually linked or related to the node. Several contents attributes can be used to overlay statistical visual feedback on the page thumbnail: $A_{12}, A_{T,6}, A_{T,7}, A_{G,6}, A_{G,7}, A_{L,6}, A_{L,7}, A_{L,8}, A_{L,9}$.
- An example of usage might consider the average of time users analyzed a webpage: if the value tends to on $\Delta t_{k_{min}}$, the page might have been used only for navigation purposes; if the value tends to $\Delta t_{k_{max}}$ the page might have been a contents store of interests for the users). This information can be easily represented using $V_{1,3}$ and $V_{1,4}$ visual element. Shape thickness or color coding can be applied to produce valuable insights.

3.2 Information Layers

The attributes sets are visually represented as needed. They are not necessarily used simultaneously on one visualization scheme but are useful for a tightly coupled

representation and interpretation between different visualization scenarios. For this purpose, visual features are classified in several classes of representation, called *Information Layers (IL)*. An Information Layer is a logical collection of web data abstractions that can be laid over a visual representation [Chen2004]. This approach was inspired from GIS (Geographical Information Systems) [GIS2005] where several map layers are manipulated to attain visual representations of the information.

To visually represent the $A \cup A_T \cup A_L \cup A_G$ attributes sets we divided the visual workspace in several information layers, using a similar approach as introduced by Chen [Chen2004]. However, some of our information layers consist of distinct visual features represented using transparency. These information layers are achieved using visual cues such as color, size, shape and thickness to code information from specific attributes sets. We can exemplify with the usage of color, transparency and thickness of link elements to represent connectivity information $V_{1,4}$; in addition, the same cues are suitable to represent visual paths followed by users overlaid on the *visual workspace* of the website (visual workspace = the information layer that combines all visual scenarios of the website's UI, mapped on the same viewport as used for analysis purposes).

Our work aims not only to make a representation of the website structure (or portions of it), as introduced by Chen [Chen2004] with the *Web Image*, but also for several representations of website's interface design layouts as visual workspace, visual representation of concurrent website usage sessions virtually mapped over the website structure, and dynamic goal-oriented interconnections of website pages. Multi-tier representations can be achieved by combining statistical information, extracted from the extended attributes sets, with structural representations that make use of structural attributes.

Each IL user a subset of the structural and extended attributes sets: $A_{ILj} \in A \cup A_T \cup A_L \cup A_G$ and depends on a subset of the complete set of visual translation functions $M_f(V_f, \phi)$.

By using the information that can be extracted from the website structure, its classified contents and usage information and combining most of the structural and extended attributes sets, several information layers can be considered for our case:

1. *Reference Map (RM)* or *Base Image (BI)* – is the layer that uses structural attributes to visually map a possible representation of the Website Structure (WS), page design layouts in a Visual Workspace (VW), etc. It makes use of shape, color, thickness and transparency to code $A_1, A_2, A_3, A_6, A_7, A_{T,1}, A_{T,2}, A_{T,3}, A_{T,5}, A_{L,1}, A_{L,2}, A_{L,3}, A_{L,4}, A_{L,5}, A_{G,1}, A_{G,2}, A_{G,3}, A_{G,4}, A_{G,5}$ attributes sets. BI is usually the background layer for most of our visual representations: Figure 3;

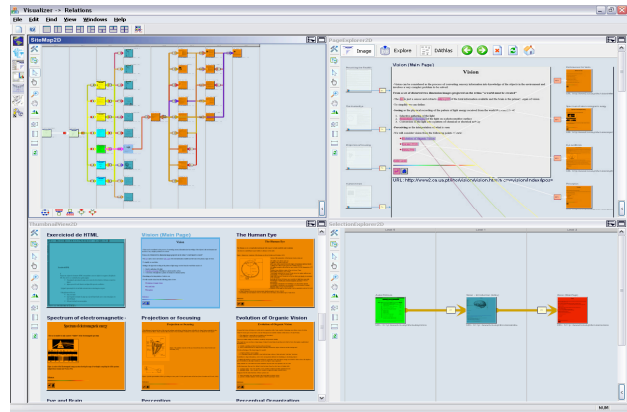


Figure 3 Reference Maps with additional attributes

2. *Number of Visits (NoV)* – represents the layers class that displays statistical information for page visits and link visits Figure 4. It uses the following extended attributes: $A_8, A_9, A_{L,8}$. Object shape, thickness, color and transparency visual clues are used to highlight page visits and link visits information. These visual clues are usually mapped on top of the *Base Image* layer – BI, using A_2, A_3 and A_6 attributes to represent it;

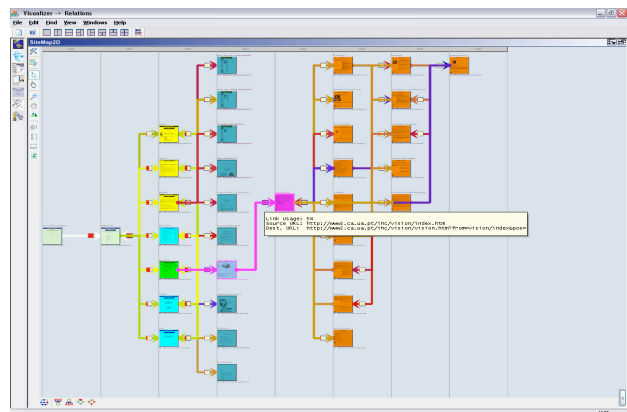


Figure 4 Visits information

3. *Links Usage (LU)* – represents the layers class that codes usage information for inter-page connection elements Figure 5. This layer visually represents the $A_{L,8}$ attribute and uses object shape, thickness, color and transparency as visual clues. It can be overlaid on top of a BI composed of the website structure as links elements between website pages, a BI that represents the visual workspace of the website as positional coded visual clues, or a BI that represents webpage contents as statistically coded hotspots information;

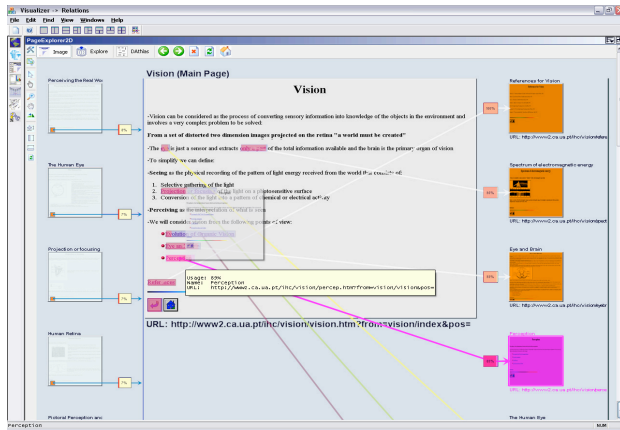


Figure 5 Hovered additional link information

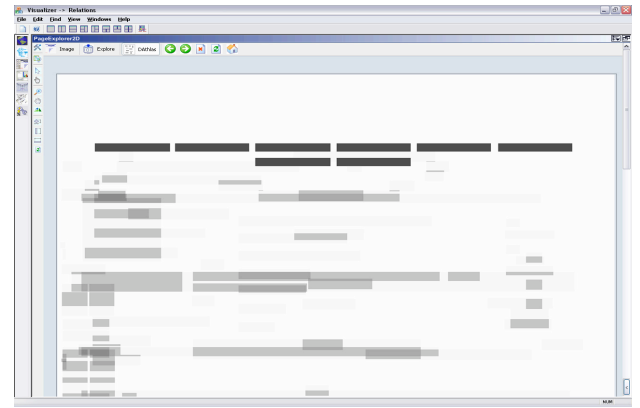


Figure 6 Visual Representations (I)

4. *Viewing Time (VT)* – is one of the most complex layers class and uses the information collected from eye / motion tracking or interception systems to represent $A_{10}, A_{11}, A_{12}, A_{T,6}, A_{T,7}, A_{L,6}, A_{L,7}, A_{L,9}, A_{G,6}, A_{G,6}$ attributes sets. The usage of visual clues to code these attributes depends of the type of information to be represented, e.g. thickness, direction, transparency and color can be used to represent $A_{L,8}$ mapped on top of the BI described by $A_{L,2}$. Several attributes as $\Delta t_{k \min}, \Delta t_{k \max}, \Delta T_{kta}, \Delta T_{kla}, \Delta T_{klj}, \Delta T_{kga}$ can be used for the VT IL, and then represented on top of an adequate BI;
5. *Inspection Density Map (IDM)* – represents the layer that uses shape and color to represent statistically derived information based on $A_{T,3}, A_{G,2}, A_{L,2}$ structural attributes – inspection and interaction areas. This IL is usually represented on top of the visual workspace BI and codes statistical usage information identified by $A_{12}, A_{T,6}, A_{T,7}, A_{G,6}, A_{G,7}, A_{L,9}, A_{L,6}, A_{L,7}$ and $A_{L,8}$ extended attributes.

These five classes of information layers might depend on the actual system implementation. Some might support dynamic filtering and interaction mechanisms while others are only supposed to represent a static view of the information. However, several derivations of these might consider additional information and / or inspection mechanisms.

The combination of a Reference Map – RM (BI) with one or several Information layers – IL produces a *Visual Representation (VR)* Figure 6 and Figure 7, as a final target for an analysis and visualization system:

$$(14) VR_f = RM_m \cup \sum_{i=1}^n IL_i, \text{ where } VR_f \text{ is a specific}$$

VR to be achieved, RM_m is the reference map (base image) used as background for the representation, and each IL_i is a specific information layer obtained from the combination of on or several extended attributes and visual translation functions $M_f(V_f, \phi)$.

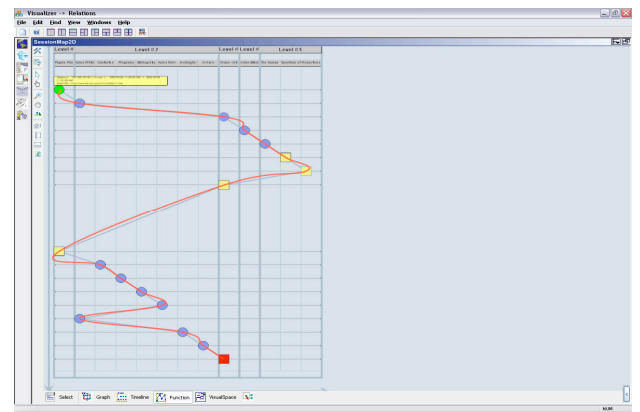


Figure 7 Visual Representations (II)

3.3 IL operators and operations

A possible definition of elementary sub-attributes, applied to the set of extended attributes, with each elementary sub-attribute being used to represent a set of atomic values in a specific active domain, can be:

$$(15) A_w = \sum_{i=1}^n a_{w,i} \mid a_{w,i} \rightarrow \lambda_{w,k} \in \nabla_{w,i} \text{ where each}$$

elementary sub-attribute $a_{w,i}$ can have a $\lambda_{w,k}$ value, defined on a specific active domain of atomic values $\nabla_{w,i}$, within the range of possible values $[\delta_{\min}, \delta_{\max}] : \delta_{\min} \leq \lambda_{w,k} \leq \delta_{\max}$.

In following paragraphs we make use of definition (15) to explain how each information layer depends on the domain of values $\nabla_{w,i}$ for each elementary attribute mapped into a visual feature.

Note that the range of possible values $[\delta_{\min}, \delta_{\max}]$ for each $a_{w,i}$ can be used as parameters on visual mapping functions ϕ to control the results for the visual representation, e.g. remapping a visual feature to a distinct domain of value might produce different results that show

or hide subsets of values from the represented domain. For this purpose, we can define ϕ as follows:

(16) $\phi_{f,w} \rightarrow \phi_{f,w}(\lambda_{w,k}, \delta_{\min}, \delta_{\max})$, where each trans-
 lation function $\phi_{f,w}$ depends of at least three parameters:
 $(\lambda_{w,k}, \delta_{\min}, \delta_{\max})$.

A complex set of parameters can be used to filter information, usually called *threshold* Figure 8. The specific set of parameters depends on the representation of each elementary attribute.

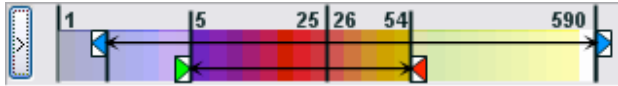


Figure 8 Example of Threshold Limits

Similar to the *Web Graph Algebra* introduced by Chen [Chen2004] and inspired by *Map Algebra* [Tomlin1994], we introduced a set of operators to manipulate information layers. These operators allow web usage mining and data operations on information layers. Unary operators are used to filter information while binary operators are used to combine the information on two distinct layers and obtain a new layer. Applying successive operators on information layers might produce distinct results, depending on what operators are used and on the intended results.

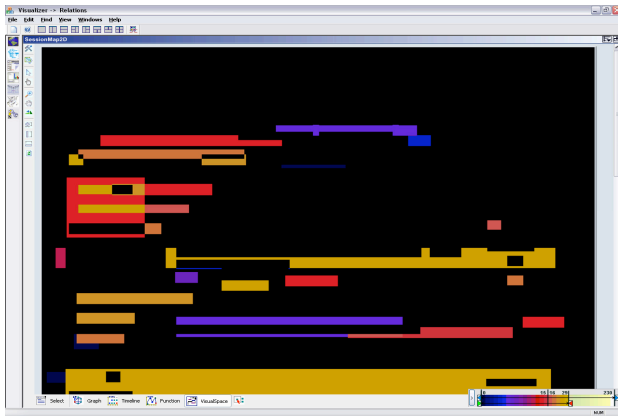


Figure 9 Modified Threshold Results

Operator FILTER is a unary operator that manipulates one information layer according to a specific threshold and filters information not in the specific domain of values specified by threshold parameters:

(17) $IL_{\theta} = FLT_{threshold}(IL_{\alpha})$, where IL_{α} is filtered
 using the threshold parameters and the resulted content is
 placed into a new IL_{θ} ;

Operator ADD is a binary operator that adds two distinct layers of information:

(18) $IL_{\theta} = IL_{\alpha} + IL_{\beta}$ selects the objects that exist in IL_{α} and IL_{β} , and transposes them into IL_{θ} . If the objects exist in both layers, the resulted value is the sum of their respective content values from both layers, otherwise is the value associated to the object; e.g. considering the object O_x present in two distinct layers, with two distinct numeric values associated to each layer, x, y , the result of the ADD operator represents the object O_x with a numeric value associated, calculated as the sum of $x + y$;

Operator MINUS is a binary operator that makes the difference of two distinct layers:

(19) $IL_{\theta} = IL_{\alpha} - IL_{\beta}$ selects the objects that exist in IL_{α} and IL_{β} , and transposes their difference into IL_{θ} .

If the objects exist in both layers, the resulted value is the difference between their respective content values from both layers, otherwise the object in the second layer is not considered. Negative values can be obtained with MINUS operator, but these values can be either considered zero or differently coded in the representation;

Operator COMMON is a binary operator that makes the intersection of two distinct layers:

(20) $IL_{\theta} = IL_{\alpha} :: IL_{\beta}$ selects the objects that exist in both IL_{α} and IL_{β} , and transposes their intersection into IL_{θ} with the respective content values available in both layers only. The resulted values can be considered: the minimum, the average or the maximum values of the intersected content in two layers;

Operator EXCEPT is a binary operator that selects the objects in only one of two distinct layers:

(21) $IL_{\theta} = IL_{\alpha} \ominus IL_{\beta}$ selects the objects that exist only in IL_{α} and do not exist IL_{β} , and transposes their content values into IL_{θ} ;

The feature that allows dynamically applying of filters and interaction with the representation domains of atomic values is designated as *Dynamic Visual Inspection and Manipulation (DVIM)*. Several parameters can be manipulated to control the behavior of visual mapping functions $M_f(V_f, \phi)$.

Example: To exemplify a possible usage of operators, we can consider a simple visualization method used to make a visual inspection of interface design (visual workspace) coherence [Mealha2004], presented in Figure 10.

This visualization methods uses attribute mapping on the original 2D visual workspace, where pages were viewed. It needs to map $A_{L,2}$ attribute of link elements as visual elements transposed as their real geometrical active area on the 2D visual workspace. In this case, the information

is presented to the viewer after being transformed with “raw information → visual feature” function $M_i(V_i, \phi_i)$. This visualization scheme can be decomposed in 2 basic visual elements $V_i = \{V_{i,1}, V_{i,2}\}$ with:

- $V_{i,1}$ = geometrical area of attribute $A_{L,2}$ that varies in color according to “hot-spot usage” calculated from attribute $A_{L,8}$, in the range that goes from δ_{\min} to δ_{\max} according to the selected filtering threshold, with $\phi_{i,1}$: that transforms the usage statistics with a linear or logarithmic function into a color for the geometrical area. For this particular example, one considered all areas as pixels, the usage associated to each pixel is calculated as sum of the usage values for all hotspots that contain the pixel. $\phi_{i,1}$ can receive as parameters the limits δ_{\min} and δ_{\max} to remap the visual outputs according to the selected values. δ_{\min} and δ_{\max} represent the minimum respectively the maximum values the $A_{L,8}$ can have for the considered set of hotspots;
- $V_{i,2}$ = connecting line between two visual representations of $A_{L,2}$ attributes, reference and the referenced page (two hot-spot subsequently clicked during a usage session), $\phi_{i,2}$: draws a line connecting two visual representations of $A_{L,2}$ attributes using three visual clues to represent $A_{L,1}, A_{L,4}$ structural attributes and $A_{L,6}, A_{L,7}, A_{L,8}$ extended attributes: direction to code $A_{L,1}, A_{L,4}$, thickness to code $A_{L,8}$, and transparency to code $A_{L,6}$ and $A_{L,7}$. $\phi_{i,2}$ can receive as parameters the limits δ_{\min} and δ_{\max} to remap the visual outputs according to the selected values.

$V_{i,1}$ represents the Reference Map (RM) layer and, at the same time, Inspection Density Map (IDM) layer, while $V_{i,2}$ is the additional information layer that combines features of Links Usage (LU) and Viewing Time (VT) layers.

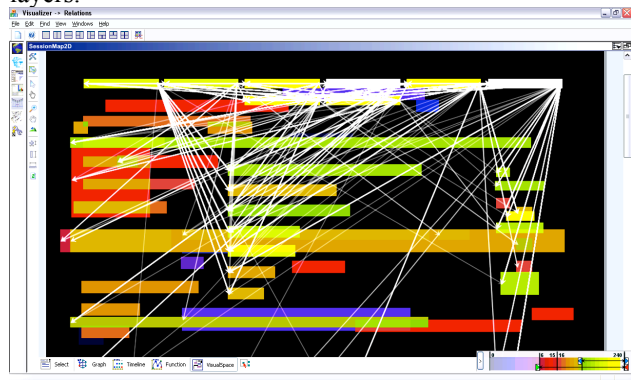


Figure 10 Visual inspection of interface design (visual workspace) coherence

First step is to use **FILTER** operator for each IL with the threshold specified by δ_{\min} and δ_{\max} parameters. The goal is to suppress the objects not in the specified usage interval. The second step is to use **ADD** operator for the two layers to obtain the representation by adding the objects in $V_{i,1}$ and $V_{i,2}$ to the representation. Note that the result depends of the threshold parameters applied for **FILTER**.

4. CONCLUSIONS

This paper contains a contribution for any researcher in the field of information visualization in the context of web based systems. The major theoretical dimension of the paper describes information attributes, information representation schemes and the possible information layers that can be used. The characterization of the specific attributes is contextualized in hypermedia systems, specifically, web information and technologies. The information visualization schemes and layers are based on research work and also have as reference many other authors of this research field.

This work represents the conceptual core for a research framework that also contains an experimental web log based management system in its second version. Some of this system’s usage screenshots are used to clarify the need and/or purpose of some attributes and operators.

Several combinations of attributes, operators, operations and visual transformations of raw information into visual features have been already tested and good feedback has been collected during experimental usage scenarios.

The focus here is to combine the definitions introduced by this work to formally describe effective inspection instruments for the analysis of generic website structure, contents and usage patterns.

5. BIBLIOGRAPHY

[Card2001] Card K. Stuart, Pirolli, P., Van Der Wege, M., Morrison, J. B., Reeder, R. W., Schraedley, P. K., and Boshart, J. 2001. Information scent as a driver of Web Behavior Graphs - results of a protocol analysis method for web usability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, United States). CHI '01. ACM Press, New York, NY, 498-505.

[Chen2004] Chen Jiyang, Sun, L., Zaiane, O. R., and Goebel, R. 2004. Visualizing and Discovering Web Navigational Patterns. In *Proceedings of the 7th international Workshop on the Web and Databases: Colocated with ACM SIGMOD/PODS 2004* (Paris, France, June 17 - 18, 2004). WebDB '04, vol. 67. ACM Press, New York, NY, 13-18.

[Drott1998] Drott M. Carl. 1998. Using web server logs to improve site design. In *Proceedings of the 16th Annual international Conference on Computer Documentation*

(Quebec, Quebec, Canada, September 24 - 26, 1998). SIGDOC '98. ACM Press, New York, NY, 43-50.

[Faraday2000] Faraday Pete. 2000. Visually Critiquing web Pages. *Proceedings of HFWeb '00* (Austin, TX, June 2000). Online at: <http://www.tri.sbc.com/hfweb/faraday/faraday.htm>.

[Fraternali2003] Fraternali Piero, Maristella Matera, Andrea Maurino. 2003. Conceptual-level log analysis for the evaluation of web application quality. *First Latin American Web Congress (LA-WEB'03)*, pp 46.

[GIS2005] Geographic Information Systems. 2005. U.S. Geographical Survey. http://erg.usgs.gov/isb/pubs/gis_poster/, Last visit: May 2006.

[Healey2001] Christopher G. Healey, Robert St. Amant, and Jiae Chang. 2001. Assisted Visualization of E-commerce Auction Agent, *presented at Graphics Interface 2001 – Canadian Human-Computer Communications Society*, pp. 201-208, 7-9 June 2001.

[IFABC2006] IFABC Global Web Standards. <http://www.ifabc.org/standards.htm>. Last visit: March 2006.

[Ivory2002] Ivory Melody Y., Marti A. Hearst. 2002. Improving website design. *IEEE Internet Computing*, v.6 n.2, p.56-63, March 2002.

[Mealha2004] Óscar Mealha, Beatriz Sousa Santos, Nunes José, Florin Zamfir. 2004. Integrated Visualization Schemes for an Information and Communication Web Log Based Management System. *Proceedings of International Conference of Information Visualization – IV04*, London. July 2004.

[Tomlin1994] Tomlin C. Map algebra-one perspective. *Landscape and Urban Planning*, pp. 30(1-2):3–12, October 1994.

[Zamfir2004] Zamfir F., Nunes J., Teixeira L., Mealha Ó., Sousa-Santos B., 2004. Visual Application for Management of Web-Based Communication and Information Systems. *Proceedings of IADIS International Conference Applied Computing 2004*, pp. II 119–125. Lisbon, Portugal.

[WCTD2006] Web Characterization Terminology & Definitions Sheet - *World Wide Web Consortium*. <http://www.w3.org/1999/05/WCA-terms/>. Last visit: March 2006.

Bridging the Gap Between Real and Electronic Documents

Renata Marin

Tiago Guerreiro

Daniel Gonçalves

Joaquim A. Jorge

Dep. Eng^a. Informática, IST

Av. Rovisco Pais, 1000 Lisboa

renata.marin@gmail.com, {tjvg,daniel.goncalves,jaj}@inesc-id.pt

Abstract

In the area of Personal Information Management, researchers try to cope with problems arising from the large amount of personal electronic information users have to deal with nowadays. Part of that information are the users' documents. It is difficult to retrieve documents that have been written weeks or months ago, as existing systems provide little or no clues to their whereabouts. However, it is often the case when a printed version of a document is available, and the users want to find the corresponding electronic document. That printed version should be enough to make the retrieval possible.

We describe how RFID tags were used to solve that problem, by allowing an automatic association between electronic and paper versions of the same document. Our solution is part of the Quill system, a narrative-based document retrieval interface. Quill allows stories about documents to be told and used to find the documents they describe. With the use of RFID technology, it is now possible to mention information about real-world documents in those stories. Furthermore, Quill maintains a knowledge base in which information about the users and their activities is stored. The synergies between that information and the one collected with the help of RFID tags allow new use scenarios to be supported.

Keywords

RFID, document retrieval, narrative-based user interfaces, personal information management.

1. INTRODUCTION

Retrieving electronic documents is not an easy task. The hierarchic nature of common filesystems does not properly reflect how users remember their documents. To store a document in a filesystem it must first be classified into the hierarchy. This leads to cognitive problems, since the users are aware that from a good classification depends an easy retrieval at a later time.

Unfortunately, the classification task is fraught with problems. It might not be evident in which category to place a document. More than one might seem adequate (how to choose?) or none might. In the latter case, should a new category be created? If so, their number will grow, making the classification of future documents increasingly difficult. If not, the document will be placed into some kind of "miscellaneous" category, almost the same as not having been classified at all. Furthermore, even if users choose what seems to be the best classification at a given point in time, it is not certain that classification will be remembered later on, or even if it would still seem the correct choice, given that new documents might influence the classification criteria. These problems are so serious that several studies have shown that users will preferably keep their documents in unsorted "piles", rather than in organized "files", relying on clues such as special memory to find them [Malone83][Rodden99].

To make matters worse, computers, in their different forms, are pervasive in our society. Most users now have access to more than one machine. The computers at home and at the office are an example of this, but also are mobile phones or PDAs, that nowadays allow users to perform several tasks that before required a "traditional" machine. In particular, personal information and documents are now created, edited and stored in those devices. The problem of finding a document became more than just looking for it in a computer's filesystem. It now requires users to remember *in which computer* it might reside [Nielsen02].

The retrieval problems we just described arise, to a large extent, from the fact that while the computer forces users to think in terms of *files*, they are indeed dealing with *documents*. The filesystem and files are metaphors created to cope with the fairly basic storage capabilities of early computing systems. Users, on the other hand, remember their documents not only in relation to an *ad-hoc* category or a filename, but instead reporting to the wider context in which those documents were handled. As mentioned by Jeff Raskin, the content of a document is its best filename [Raskin00]. Users remember what is in their documents, when they were handled or why they were read or written. Using this autobiographic information to describe and retrieve documents would be helpful, as it mimics how users tend to remember them.

Whittaker's work confirms this, showing that many users resort to their email clients to store documents, sometimes inside "fake" email messages [Whittaker96]. Even if those tools don't directly support such tasks, all messages have associated to them information that will make the retrieval of documents easier: a subject, a date, etc.

While using autobiographic information to retrieve documents is a promising idea, just asking users for that information would yield poor results. It is necessary to somehow elicit it from them, and for the computer to understand it. To solve that problem we developed a narrative-based interface for document retrieval. Users will tell stories describing the documents they seek, and those stories will be used to identify them. All humans tell stories all their lives, from an early age. This is an innate human ability, as it was found that across cultures, even those of remote tribes, stories are told [Brown91]. It is, thus, a natural way for users to express themselves. Furthermore, the different story elements that compose a narrative appear not isolated from each other, but integrated into a coherent whole. Telling a story will enable users to more easily recall relevant information, instead of just trying to remember independent factoids about a document.

While developing Quill, our narrative-based document retrieval system, and choosing the shape its interface would take (described in Section 4), we identified the different elements that might appear in document-describing stories. Among them are references to related documents and to the document itself in printed form. It soon became evident that some way of taking printed documents into account when trying to retrieve a document was necessary. The work we present in this paper will complement Quill, establishing a relationship between virtual documents and their real world replicas.

We resorted to RFID technology to bridge the gap between electronic and paper documents. RFID tags are very cheap. RFID readers are also becoming cheaper and smaller. Thus, this technology can be used to identify the different paper documents handled by users with little cost. Our approach consists on the semi-automatic association of paper documents to their electronic counterparts whenever they are printed, with the help of an RFID tag. This allows users to employ one to find the other. Furthermore, by integrating this feature into Quill, data about real documents can be considered at the same time as other autobiographic information about the users, supporting different interaction scenarios rather than just allowing the blind retrieval of a document.

In the following section we will describe other approaches that try to use real documents for information management and retrieval. After identifying their shortcomings, we'll describe what a narrative-based interface is, and how such an interface can be used for document retrieval. This will lead us to a description of the Quill system, into which our solution was integrated. We'll then present our approach, detailing the different supported interaction scenarios, and mention some prelimi-

nary results. Finally, we'll conclude pointing to interesting possible future work.

2. RELATED WORK

As the sizes of our Personal Document Spaces increased, it became a priority to improve and ease the document-retrieval process. However, even with fast virtual document retrieval techniques, there is still a gap between the virtual space and the real printed documents. In our everyday life we can easily find situations like this:

"Isn't that the paper the teacher talked about yesterday? Can you send it to me, please?" asks Michael.

"Sure", answers Charles.

Half an hour later, Charles arrives at home and wants to send the pdf version of the paper to Michael. But ... Where is the file? Charles has the printed version but doesn't find the pdf it comes from.

"- Dammed!" says Charles, -"I've got here the printed version but I can't do anything with it..."

The problem here is that while both documents, real and virtual, are the same, when the pdf file was printed the connection between them was lost. They still are just different facets of the same document, albeit in different media. As such, we should strive to maintain a relationship between them.

Some previous works try to accomplish this in different ways. *Video-Based Document Tracking [Kim04]* is an interesting work that tries to solve the physical/virtual document association problem. In the solution it proposes, an overhead camera is used to track physical documents on a desktop and link them to the corresponding virtual documents. The system detects changes in a document stack, and uses this information to establish the desk's contents. The movement of documents is tracked with a video-camera. The video is analyzed with computer vision techniques to connect the document with his virtual copy on the disk. Although this approach can relate some documents with their virtual replicas it isn't as very accurate, as should happen in a retrieval system. It relies on the visual pattern of the first page of a document to identify it, which is clearly not enough when several documents with similar first pages are present at the same time. Also this technique can only recognize documents if their first page is facing upwards and not occluded by some other document or object.

Other solutions, aiming at a more accurate linking between virtual and real documents are based on RFID (Radio Frequency Identification) technology.

Historically, the roots of the RFID technology can be traced back to the World War II. It was used by the British to distinguish their aircrafts from the enemy's. RFID is a generic term for technologies that use waves to automatically identify objects. Each tag is identified with a unique serial number. A microchip is attached to an antenna that enables the RFID to transmit the identification

to a reader. This reader converts the radio waves reflected back from the RFID tag into digital information that can be passed on to computers [RFID].

One of the earliest works that tried to bridge physical and virtual worlds through the use of RFID technology was *Want et al.* [Want99]. It tries to connect the physical objects with its virtual representation using various types of tags.

More recently, the bridge between virtual representations and real objects has been extended to personal items that accompany the users on their everyday tasks [Borriello04]. This work tries to help people not to forget important objects. The different relevant objects are tagged. RFID readers are present at each of the locations where the user usually spends some time at. Users are given a mini personal server that communicates with their watches. Whenever a user passes close to a reader without an important object (that he could have forgotten) the watch reminds the user of it. This work shows how RFID technology can be employed not only to help users to initiate the retrieval of objects they remember, but also how the computer itself can identify a user's needs and act proactively to provide the required objects.

Abu Safiya et al. [AbuSafiya04] developed a project that more directly addresses the information retrieval problem. The concept of Document Database (where all electronic documents are indexed) is extended to allow the representation of printed documents and their physical location. With this, a company can manage all its documents, becoming possible to know at all times where a specific document is or which documents are in a specific room. The RFID readers must be located in strategic positions, places where documents usually accumulate. While interesting, this work is a large scale project, trying to deal with an organization's entire document collection. Thus, only important documents are tagged identified. It is mostly a localization-based project and the retrieved information is quite poor. Also, the retrieval process is unidirectional: a user can't retrieve a virtual document from its real replica.

Our goal is similar to that of the systems we just described: to use RFID technology to manage and relate virtual and real documents uniformly and efficiently. However, our approach strictly focuses on personal document spaces having in mind the retrieval of virtual documents from their real-world counterparts and vice-versa. Since we're trying to help users manage their personal documents, our solution, unlike those above, instead of simply maintaining the link between real and virtual documents, uses a wealth of autobiographic information to enrich the documents' descriptions and further enhance the users' ability to find them.

3. NARRATIVE-BASED INTERFACES

Previous studies had shown that autobiographic information might be useful to organize and retrieve documents. The use of narratives as a way for users to convey that information to the computer seemed promising. Thus, we

began to design a narrative-based interface for document retrieval. Rather than using stories to annotate documents, something few users would even consider doing in a consistent way for all their documents, we focused on using narratives as a way to describe documents at retrieval time.

To validate the approach, we conducted a set of interviews in which users were asked to tell stories about their documents [Gonçalves03] [Gonçalves04]. We collected 60 such stories, told by 20 different users. Each user told a story about a Recent document (written by the user less than two weeks earlier), an Old document (written by the user more than 6 months ago), and a document with Other authors. The interviews were recorded, transcribed, and contents and relational analysis were performed on the transcripts [Huberman91].

Overall, we found stories to be composed of 17 different elements: Time, Place, Co-Authors, Purpose, Author, Subject, Other Documents, Personal Life Events, World in General, Exchanges, Type, Tasks, Storage Location, Versions, Contents, Events occurring when handling the document, and the document's Name. Each element occurs as a semantically significant sentence or phrase in the stories.

We tried to find differences in stories arising from (among other factors) the different document types, the elapsed time since the document had been handled and user gender and age. No noteworthy differences were found, with one exception: the length of stories about documents written by the users is slightly greater than that of those they didn't write (17 vs. 11 elements, on average). The different story elements also appeared with different frequencies. Most notably, Place, Co-Authors, Purpose, Author, and Versions appear less frequently. Statistical tests (with 95% confidence) confirmed these results.

In short, it was confirmed that narratives about documents convey lots of information about those documents, and can be easily told by all, regardless of gender and age. We also got an extensive description of what information to expect in stories, and in what order.

Based on those results, we were able to create some guidelines for the design of narrative-based document retrieval interfaces. The most important are:

- In general, no user customization will be necessary in relation to what to expect from a story.
- It is important to determine the kind of document being described early in the narrative, to correctly form expectations about what can be found ahead in the story.
- The typical story structures we found should be used to help understand what element is being described by the users at any given time, and to help understand the information therein.
- Users tend to digress. As such, it is important to establish dialogues with them in order to keep them on

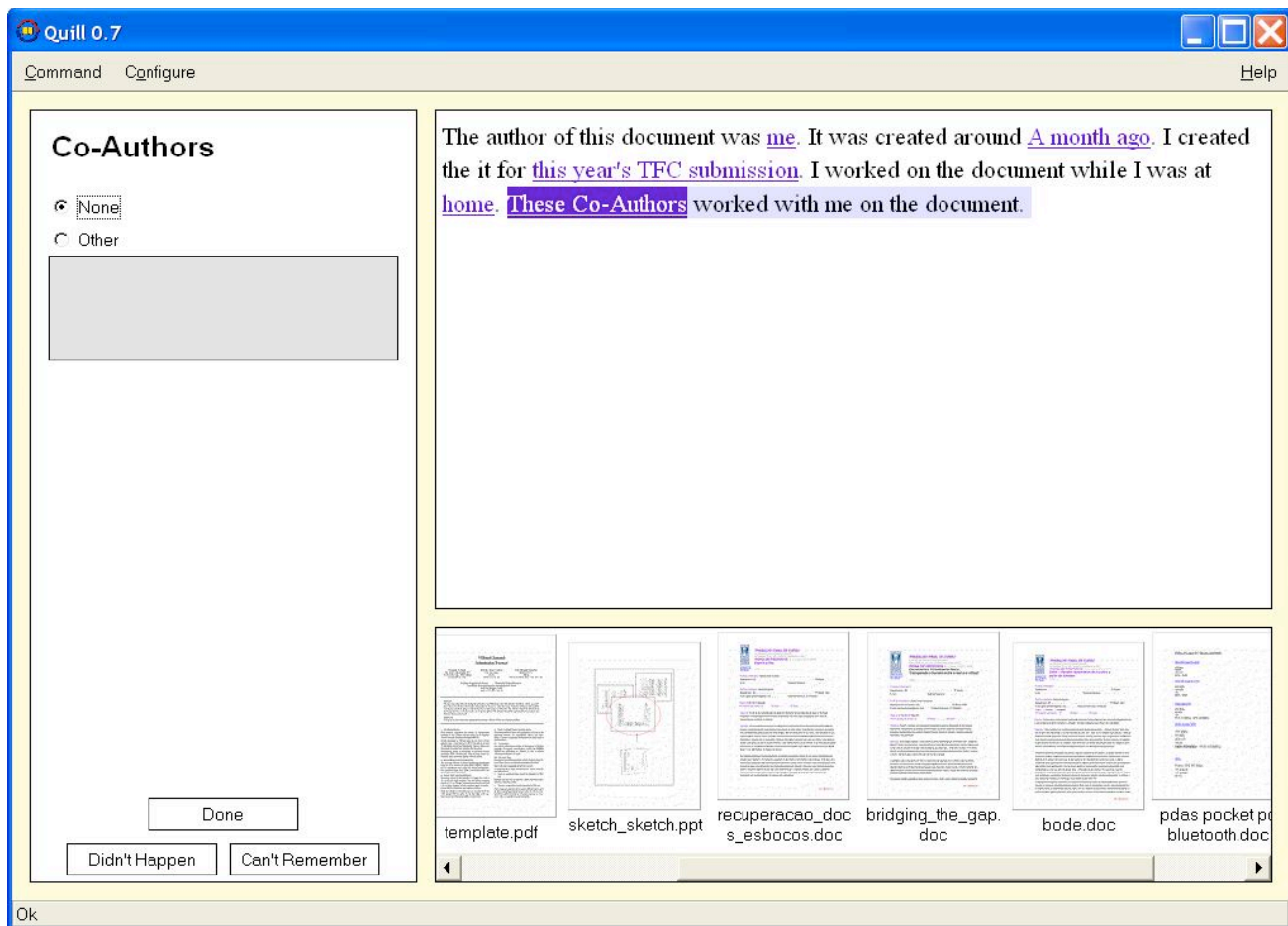


Figure 1 – The Quill Interface

track and to obtain all information they can actually remember.

- We found that the users' visual and special memory play important roles in stories. Some technique that identifies the overall structure or visual appearance of a document and can use that information to differentiate among several documents would be useful.
- Related documents are often mentioned in "stories within the story". The interface should be able to capture those stories and, at the same time, lead the users back to the description of the target document.
- Actions such as printing a document are mentioned fairly often. We should take those actions into account. Printing and printed documents, in particular, should be taken into consideration.

In possession of these guidelines, we knew what to expect from stories, and had an idea of the features the interface to which those stories are told should take. The actual shape of that interface remained to be established.

To do so, we created and evaluated two low-fidelity prototypes of narrative-based interfaces for document retrieval [Gonçalves04a]. Both embodied the design guidelines. The first was based on the direct manipulation of the different story elements, depicted as boxes that could be arranged on screen to create the stories. The second

represented the story as text. Each element corresponded to a sentence. Incomplete sentences were presented to the user that only had to fill in the missing information. In both prototypes the story elements were entered with the help of specialized dialogues, and suggested to the user in the order found to be the most likely in the interviews. In an actual system, those stories would then be used to identify the document being sought by the users.

Again, we collected 60 stories, from 20 users, using both prototypes. The second prototype was undoubtedly the better one. Stories told using it were similar to those told to humans, both in terms of content and structure. The users' subjective satisfaction was also clearly better for that prototype.

Based on these results, we implemented the Quill interface, described below. Quill allows the users to tell their stories and uses them to identify, from an index, the document that best matches it. With Quill, we performed a third and final study regarding the validity of stories as a way to convey information about documents. In that study, the information in stories was evaluated regarding its accuracy [Gonçalves05]. We found that on average, we can expect between 81% and 91% of the information in each story to be correct. This means that only between 1 and 3 story elements might be wrong (out of 17). Furthermore, we found that, for the most part, the informa-

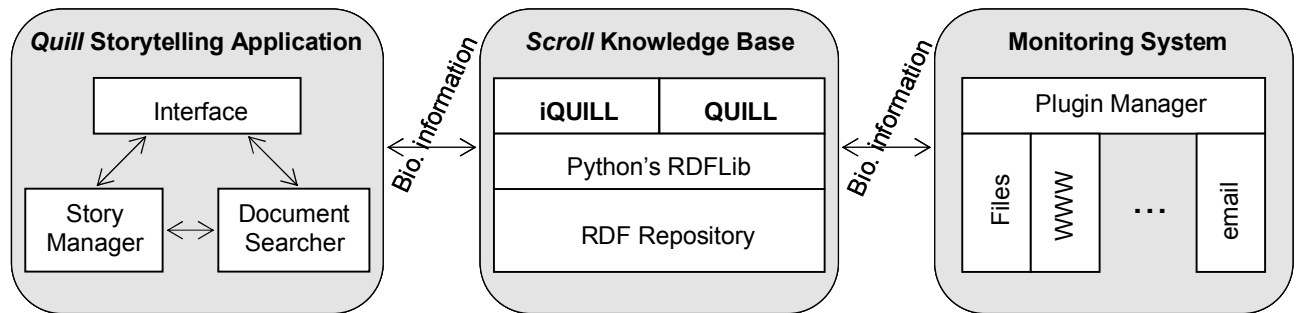


Figure 2 – The Quill System Architecture

tion isn't completely incorrect. For instance, when mentioning a document's name, only part of it is usually wrong, or the words in the name might be mentioned in an erroneous order ("report200" instead of "2000report", for instance). Thus, if the information is not taken as an absolute truth, it might still help identify the desired document. Surprisingly, stories about Recent and Old documents are equally accurate. The only thing that varies is the precision with which the different elements are mentioned. For instance, we find time ranges of a month instead of a few days, but correct nevertheless.

Narrative-based interfaces were, thus, confirmed as being a valid and sound approach for users to convey information for document retrieval.

4. THE QUILL SYSTEM

The Quill interface is depicted in Figure 1. The story is created in the top-right area. A sentence representing each possible story element is suggested, in turn, to the user. In that sentence, the information to be completed is highlighted. The user must then enter it using the dialogue box in the right. There is a different dialogue for each story element, and we ensure the appropriate dialogue is visible. Additionally, the user can state that something didn't happen, or that he can't remember a particular story element. This is done with the help of two buttons at the bottom of the dialogue boxes. It allows the interface to tell the difference between something the users don't know, and something they know not to have happened. Also, while Quill suggests each story element to the user in the order found to be the most likely in previous studies, the user remains in control, being able to choose any element to mention at any point in time. Quill adapts to those changes over time, fine-tuning its suggestion order with the help of hidden-markov models.

The sentences are adapted whenever new information is entered by the user, to make them coherent with that information. We took care not to change them too much, to prevent confusing the users.

As the story grows, promising documents are found. A thumbnail of those documents is displayed in the bottom of the window. This takes advantage of the fact that users often remember the overall aspect of a document (visual memory plays an important role). This way of presenting the probable matches is non intrusive and does not unduly

distract the users from the storytelling process. Often, just by quickly scanning the list, it is possible to tell if the target document has been found or not.

Limited natural-language understanding is provided. A full parser tries to understand the phrases entered by the users and automatically generate their semantics. If this fails, a tag parser tries to extract as much meaning as possible from the phrases.

A sample story, collected by Quill, is presented below:

The author of this document was me. It was created around 10 of May of 2004. I created it for PCM Report. I worked on the document while I was at home and the workplace and At my colleague's home, in college. André Martins worked with me on the document. The document is about CGEMS Advanced Search Engine. This document reminds me of no other. I sent it to André Martins using email and LAN (shared folders, etc.). It's a PDF document. The document contains the words or expressions "Search Engines, CGEMS, Java, SIGGRAPH" and looks like a two-column with lots of images and a little text. The document is stored in Laptop and Other computer. To write it, I had to developed a prototype for PCM, Search the Web, Read many related papers. It had different versions. Its filename was something like "pcm final".

As can be seen, while the syntax isn't perfect, it is sufficiently good to allow it to be read and understood by the users.

4.1 Behind the Scenes

Underlying Quill is the Scroll Knowledge Base (KB). This knowledge base uses RDF as its knowledge repre-

sentation formalism, and stores knowledge about the users, their documents, and their actions (see Figure 2). A special-purpose library called *Scroll* provides high level methods to deal with the RDF store, and allows path- and node-based inference to be performed. Also, several case-frames were defined. The Quill Schema allows us to represent all knowledge mentioned above, and the iQuill Schema makes it possible to represent first-order logic-like formulae and use them to perform inferences.

Whenever new information is entered into the story, inference rules are generated and evaluated in the KB. From that evaluation, relevant documents are identified. To each is given a score. The scores from resulting from all inference rules are added and the resulting value is used to establish a document ranking. The best placed documents are suggested to the user as possible matches.

4.2 Monitoring the User

To feed knowledge into the KB we created a monitoring system. This system continuously examines what is going on in the user's computer and selects relevant knowledge to be stored in the KB. It is a plugin-based system. Currently, we are able to monitor the following:

- All documents present in the users' computer, keeping track of their whereabouts, and of when they are created, modified, copied or deleted.
- All web pages visited by the user.
- All email messages sent and received by the user, as well as the attachments therein.
- All programs the user runs.
- The user's agenda (appointments, contacts, etc.)
- Real documents (with the help of RFID tags, as described in the following section).

While independent, the different plugins generate a coherent body of knowledge. For instance, if the email plugin finds an attachment that has been saved somewhere in the hard drive (as identified by the document plugin), no new document will be recorded. Rather, it will store in the KB that the existing document was sent by email to someone, enriching the information gathered on that document.

Finally, the plugins try to extract as much information as possible from the documents. This includes meta-information stored in them (ID3 tags for music files, for instance, or META tags in HTML documents). For textual documents, relevant keywords are also extracted. First, the text is tokenized and stemmed. Then the TFIDF algorithm is used to select the most relevant keywords for each document.

5. BRIDGING THE GAP

From the user's perspective, a document is often connected with a real physical copy of itself. It became evident in our user studies that the relationship between virtual and real documents is often important in the document retrieval process. They are seen as different facets of the same document, rather than separate entities.

Hence, as the relation is bi-directional we can augment our retrieval process to find both virtual and real copies of a certain document.

To bridge this gap and relate both worlds, we built our new plugin based on an RFID infrastructure. It is composed by a fixed RFID reader (with a range of 30-40cm) that is capable of reading multiple tags simultaneously, connected to a Personal Computer. It is important that it has a relatively short range, since we want to know precisely where our documents are (a desk or a shelf, for instance). The short range prevents overlapping readings if multiple readers are to be used. Our RFID tags are passive (with no power source), as we only need to receive the tag ID. Each printed document will have its own tag, which identifies it. The plugin associates the tag's ID with the corresponding virtual document (Figure 3).

We used RFID tags rather than other ways to identify documents, most notably barcodes, because they require little attention from the user. A barcode could be automatically generated and printed with the document. With RFID tags, on the other hand, the users must attach a tag to each document they print. Also, since the computer doesn't know beforehand what tag will identify a document being printed, a semi-automatic way to perform the association between an electronic document and the tag had to be devised (described in section 5.2). Nevertheless, this occurs only once, when the document is printed. Afterwards, the users have only to place the documents wherever they choose (a desk, a cabinet, etc.), and if an RFID reader is nearby, the document's location will be known to the system. Using barcodes, each time a document is moved it must be swept by a barcode reader. This might distract the user from the current task, and is liable to be forgotten.

The trade-of between having to remember to record the document position every time its moved and having to explicitly identify it once at print time led us to opt for RFID technology.

To accomplish our goals, we dealt with two different but related problems while developing the plugin: Abstracting from the low-level communication with the RFID reader operation into meaningful, document-related tasks, and discovering which documents were printed.

5.1 Reader Data Acquisition.

The plugin periodically receives data from the reader. Whenever a tag is detected, the plugin processes it, classifying it as a new or already known tag. This process can be executed over a set of tags making it possible to detect changes in a document pile or another location where several documents are stored.

If the tag is new, we face a new paper document for the first time and must determine to which virtual document it corresponds. If not, it means that a previously known document has changed place. In both cases, the information in the KB will be updated accordingly.

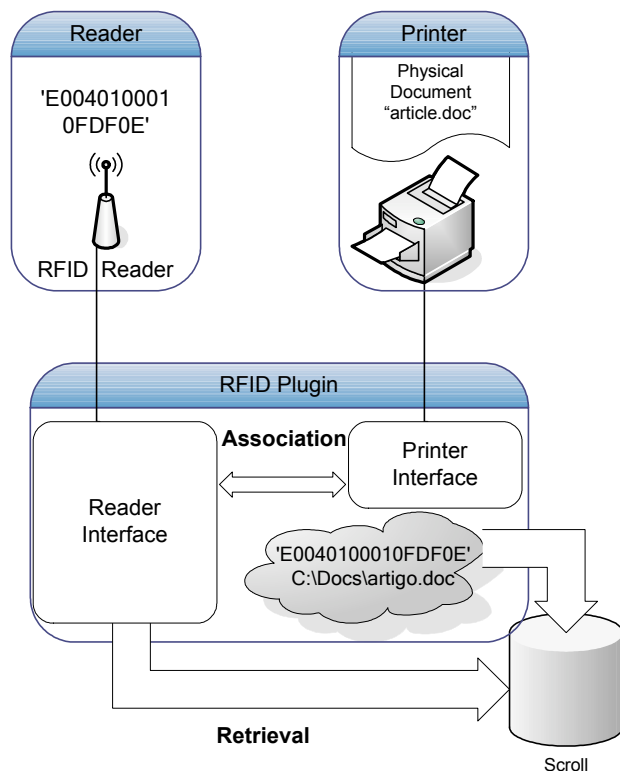


Figure 3 – Real and Virtual Document Association

5.2 Automatic documents printing detection

To associate a printed document to its virtual counterpart, we need to know when a document passes from the virtual world to the real one. By other words, when a document is printed. We assume that when a document is printed an RFID tag will be attached to it by the user, as described before. As a result, the document will be univocally identified by the hexadecimal key of the RFID. Establishing the association with a low effort from the part of the user must then be accomplished.

To accomplish this in the most automatic and less cumbersome possible manner, we follow five steps:

- Intercept the printing event;
- get the name of the file that was printed;
- read the printed document’s ID key;
- establish a relationship between the file and the key;
- store the relevant knowledge in our KB.

To detect when a document is printed, we have to capture the operating system’s printing events. This allows us to detect when a new document is being printed and needs to be associated to the respective file. We can get the file name from the print job event, but not the entire path. To determine exactly which file is being printed, we use our Scroll knowledge base, where all the users’ documents are indexed, to identify all files with that name. If there is only one such file, the association is automatic. On the other hand, if the query returns more than one result, several different files with the same filename, the users are

asked to choose between them. All different possibilities will be displayed in a list, sorted by modification date, since it is likely that a file being printed has been recently modified.

The different candidate files are shown to the user in a dialogue box that shows up when a document finishes printing (Figure 4). There, the users have the option not to associate an RFID tag to their document. If they accept, the information about the document in the knowledge base is updated with the respective RFID key. The bridge between both representations is thus created and both virtual and real copies are enriched with extra-information.

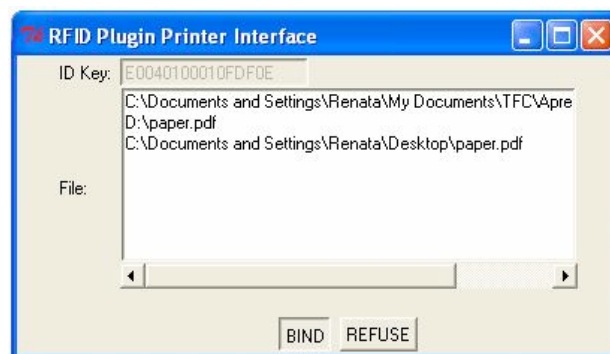


Figure 4 – Prototype of the RFID Printer Plugin

This relationship between the knowledge base representation of our virtual documents and their physical replicas enables us to retrieve documents in different ways. Considering the narrative document retrieval system, Quill, it is now possible to expand the narratives to the real documents context, information that is frequently mentioned by users. Thus, any user can mention in his story the printing occurrence, the real placement of a replica (“*and there is a real copy of the document in that shelf*”) or, with the real document in his hands, retrieve the virtual copy (“*it’s this document*”). Besides, when a document is retrieved, and we don’t know the physical copy’s whereabouts, we can ask Quill.

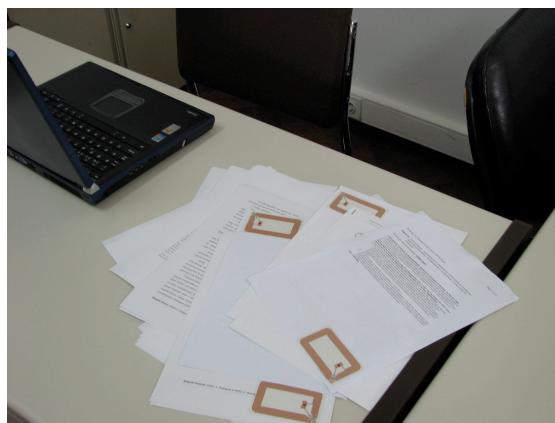


Figure 5 – Multiple Tagged Documents

Also, we developed another application that uses the knowledge base but is independent from our retrieval GUI. It is focused on the history of a real document. For example, if we have a printed document we may want to know where it came from, who did we send it to, and obviously, where is the document in our virtual space. In short, as the bidirectional bridge between both worlds (real and virtual) is established we are now able to retrieve documents, either physical or virtual, and information about them, in a direct, seamless way.

5.3 Evaluating the Association Accuracy

Our approach makes the establishing of a relationship between paper and electronic documents as seamless and effortless as possible. However, it is based on the assumption that most printed documents have a unique filename, allowing the RFID plugin to correctly guess the file being printed. To verify if this is, indeed, the case, we performed a user study in which the documents several users were considered.

A special-purpose program (for Windows machines) that traverses the users' hard drives and analyses the files therein was created. For each user, the program recorded the number of times each filename was found. The program was made available online and a request for participation was sent to our research group's mailing list. Participants needed only to download the program and run it on their machines. It would automatically look at the contents of the "My Documents" and "Desktop" folders and index them. A blacklist / whitelist mechanism (to allow other folders to be analyzed) was available and used by some participants that had most of their documents outside the default folders.

The program considered only files that were established to be documents, based on their extension. For instance, ".exe" files were not considered, but ".doc", ".pdf" or ".jpg" files were.

It was important to make the program as easy to use and understand as possible, to alleviate possible privacy concerns. Thus, its results would be written to a file in text format, so that the users could examine it before submitting the results to us by email.

Overall, 23 participants sent in their results. All participants were engineering graduate or undergraduate students, or faculty. Some ran the program on more than one machine, resulting in 27 datasets.

As can be seen in Figure 6, most (54.4%) of the 192019 files analyzed by the program have unique filenames. These will be automatically identified by our RFID plugin. For the remaining files, several candidates must be suggested to the user. However, for the most part, the number of candidates won't be superior to seven, within the number of elements that can be held and processed by the users' short-term memories. Furthermore, since those possibilities will be presented sorted by modification date, the document they seek will most likely be the first in the list. Only for 12.67% of files will there be more than 7 candidate files.

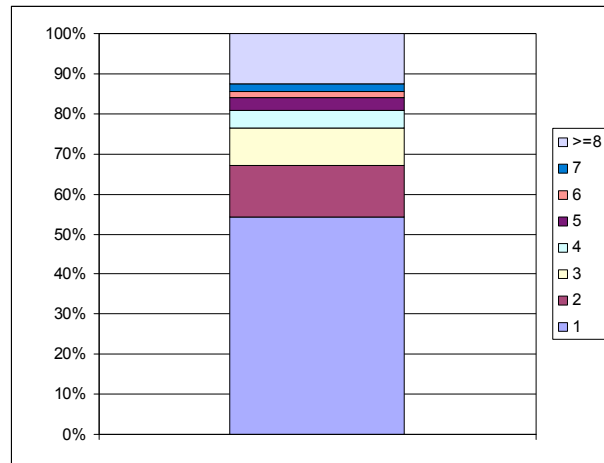


Figure 6 – % of files by filename repetition

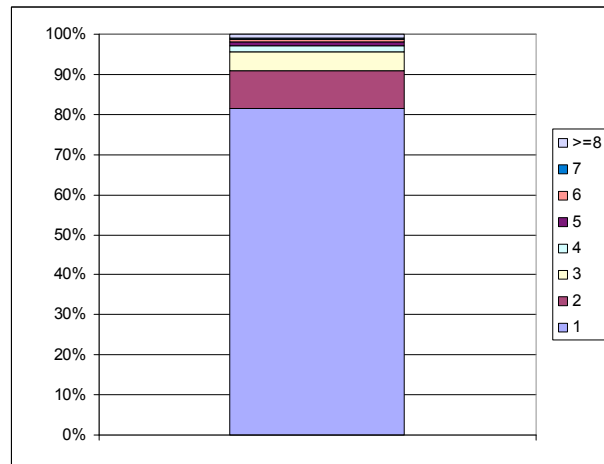


Figure 7 – % of names by filename repetition

Figure 7 shows us the percentages of repeated names, instead of files. There, we can see that 95.74% of names are used for three files at the most. Indeed, 81.42% unequivocally identify a document. This shows that the results where only the numbers of files are taken into account can be misleading. A single name occurring 465 times on a user's computer (and other similar cases) can somewhat distort the results. What is more, a closer look at those files whose names occur more than seven times shows that, for the most part, they are not documents that the users will normally print (no user would normally give the same name to 465 files...). For instance, it is often the case where they are text files accompanying software packages (the classic "readme.txt" files). For that reason, the files that users might want to print will likely share their name with up to three or four other files, well within the limit of what the users can manage.

6. CONCLUSIONS

Retrieving personal electronic documents is not an easy task. One way to make it easier is to resort not only to the hints provided by the filesystem, but also to a wealth of autobiographic information that users often remember

about their documents. References to printed documents are common. However, once a document is printed, its relationship with the electronic version is lost.

We developed a way in which it is possible to associate printed documents to their electronic counterparts with low effort from the part of the user. Our solution makes it possible to automatically detect when and what documents are printed. The identification of the printed file is not perfect, but we've shown it will automatically succeed for nearly 55% of all files, requiring little effort from the user if that is not the case. Simply by sticking an RFID tag to the printed document, the relationship is established. This supports a variety of interaction scenarios, in which not only electronic documents can be retrieved from their physical counterpart, but also where a printed copy of an electronic document can be located. Furthermore, since our solution was integrated into Quill, a narrative-based document retrieval system, the synergies between the printed-electronic document association and knowledge about the users and their activities could be explored.

In the near future, we'd like to test our approach with more than one reader, to extend the usage scenarios by allowing users to find printed documents that might be located in different places. The placement of a reader near the office door and the use of RFID tags by those who enter and leave it would extend those scenarios further, by allowing the user to know who took a document and when.

As equipment prices continue to drop, the user interaction could be made more seamless, by using printers that are able to print not only the document but also the tag, making it unnecessary for the user to stick the tags themselves.

7. ACKNOWLEDGEMENTS

The authors would like to thank all users that participated in the studies described in this paper. This work was funded by Project BIRD, FCT POSI/EIA/59022/2004.

8. REFERENCES

[AbuSafiya04] AbuSafiya, M. and Mazumdar, S.: *Accommodating Paper in Document Databases*, DocEng'04.

[Borriello04] Borriello, G., Brunette, B., Hall, M., Hartung, C., Tangney, C.: *Reminding about Tagged Objects using Passive RFIDs*, Sixth International Conference on Ubiquitous Computing, UbiComp 2004.

[Brown91] Brown, D. E.. *Human Universals*. New York: McGraw-Hill 1991. ISBN: 0-07-008209-X.

[Gonçalves03] Gonçalves, D. *Telling Stories About Documents*, Technical Report, Instituto Superior Técnico, 2003 (http://narrative.shorturl.com/files/telling_stories.zip)

[Gonçalves04] Gonçalves, D. and Jorge, J. *Telling Stories to Computers*. In Proceedings CHI2004, ACM Press, 27-29 April 2004, Vienna, Austria.

[Gonçalves04a] Gonçalves, D. and Jorge, J., "Tell Me a Story": *Issues on the Design of Document Retrieval Systems*. In Proceedings DSV-IS'04, Lecture Notes on Computer Science, Springer-Verlag, July 2004, Hamburg, Germany.

[Gonçalves05] Gonçalves, D. *Real Stories about Real Documents: Evaluating the Trustworthiness of Document-Describing Stories*. Technical Report, IST/UTL. March 2005, (at http://immi.inesc.pt/~djvg/phd/files/real_stories.zip)

[Huberman91] Huberman, M. and Miles, M. *Analyse des données qualitatives. Recueil de nouvelles méthodes*. Bruxelles, De Boeck, 1991.

[Kim04] Kim, J., M.Seitz, S., Agrawala, M.: *Video-Based Document Tracking: Unifying Your Physical and Electronic Desktops*, UIST '04.

[Malone83] Malone, T. *How do People Organize their Desks? Implications for the Design of Office Information Systems*, ACM Transactions on Office Information Systems, 1(1), pp 99-112, ACM Press 1983.

[Nielsen02] Nielsen, J. *Supporting multiple-location users*. Jakob Nielsen's Alertbox, May 26, 2002. <http://www.useit.com/alertbox/20020526.html>

[Raskin00] Jef Raskin. *The Humane Interface*. Addison-Wesley, 2000. ISBN: 0201379376.

[RFID] RFID Journal – The History of RFID Technology <http://www.rfidjournal.com/article/articleview/1338/1/129>.

[Rodden99] Rodden, K.. *How do people organise their Photographs*. In Proceedings of the BCS IRSG 21st Annual Colloquium on Information Retrieval Research. 1999.

[Want99] Want, R., Fishkin, K., Gujar, A., Harrison, B.: *Bridging Physical and Virtual Worlds with Electronic Tags*. ACM Conference on Human Factors in Computing Systems, Pittsburgh, PA, May 1999, 370-377.

[Whittaker96] Whittaker, S and Sidner, C. *Email overload exploring personal information management of email*. In Conference proceedings on Human factors in computing systems, pages 276-283. ACM Press, 1996. ISBN 0-89791-777-4.

Pesquisa de Imagens na Web: Estudo de Usabilidade

Paulo Pombinho de Matos Ana Paula Afonso
Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa
Campo Grande, 1749-016, Lisboa
ppombinho@lasige.di.fc.ul.pt, apa@di.fc.ul.pt

Sumário

Os utilizadores de sistemas de pesquisa de imagens na Web expressam as suas necessidades através de interrogações que descrevem as imagens pretendidas. A análise detalhada destas interrogações e a observação do comportamento dos utilizadores na utilização destes sistemas são aspectos fundamentais não só para a compreensão das necessidades dos utilizadores, mas também para facilitar o design de sistemas de pesquisa de imagens. A maioria dos estudos tem abordado apenas a análise de pesquisas genéricas na Web sendo escassos aqueles que observam o comportamento dos utilizadores na pesquisa de informação multimédia na Web. Este artigo descreve o estudo que envolveu a análise das interrogações e a observação dos utilizadores na utilização de sistemas de pesquisa de imagens na Web. A análise dos resultados revelou que os utilizadores preferem efectuar pesquisas textuais e utilizam 2 termos em média por pesquisa. O recurso a outros métodos de especificação da pesquisa (por esboço ou imagem exemplo) apenas é utilizado em caso de insucesso do método textual. O resultado do estudo permitiu ainda extrair as principais implicações no design de interfaces para sistemas de pesquisa de imagens na Web. Com base nestes indicadores foi realizado um protótipo da interface de pesquisa de imagens na Web.

Palavras-chave

Pesquisa de imagens na Web, Pesquisa textual e visual de imagens, Interfaces de pesquisa de imagens, Avaliação de interfaces.

1. INTRODUÇÃO

A massificação do acesso à Web na segunda metade da década de 90 e o conseqüente crescimento explosivo da informação aí disponível, em particular da informação multimédia (por exemplo: imagens e vídeo), colocou novos desafios à forma ideal de recuperar essa informação. De acordo com um estudo recente, existem 180 milhões de imagens na Web, as quais perfazem um total de cerca de 3 terabytes de dados [Goodrum00]. Paradoxalmente, este crescimento dificulta frequentemente o trabalho do utilizador, particularmente o do leigo ou ocasional, na pesquisa das imagens mais relevantes para a sua necessidade de informação [Kherfi04].

A maioria dos sistemas que focaram a pesquisa de informação multimédia fê-lo através de interfaces de pesquisa textual. Contudo, este tipo de pesquisa tem algumas limitações, uma vez que nem sempre é possível especificar uma necessidade de uma imagem em modo textual [Goodrum00; Gevers97]. Por exemplo, um utilizador que pretenda obter uma imagem de um quadro do qual não sabe nem o título, nem o autor, será muito difícil encontrá-lo através de uma pesquisa com especificação textual. Pelo contrário, uma especificação através de um esboço poderia, eventualmente, com maior facilidade, fornecer os resultados pretendidos [Holt94].

Adicionalmente, é provável que os peritos num determinado domínio pretendam uma interface gráfica sofisticada para especificar as suas interrogações. Pelo contrário, os utilizadores casuais e/ou inexperientes podem preferir especificar as suas interrogações usando a linguagem natural. É mais fácil expressar uma procura em linguagem natural, como por exemplo, “Mostra-me imagens de montanhas cobertas de neve”, do que realizar um esboço da montanha coberta de textura de neve [Gudivada95]. As funcionalidades que os utilizadores pretendem num sistema de pesquisa de imagens são influenciadas pelas suas necessidades e pelo uso a dar às imagens obtidas. Utilizadores de diferentes perfis podem necessitar de diferentes estilos de interacção [Kherfi04].

Muitos autores têm efectuado estudos de usabilidade a utilizadores de sistemas de pesquisa de imagens não especializados num determinado domínio, quer em colecções não digitais [Hastings95; Keister94], quer em colecções digitais [Lawrence99; Gudivada95; Turner95]. Contudo, poucos estudos foram conduzidos com o objectivo de analisar os utilizadores de sistemas de pesquisa de imagens na Web. Os autores Smith e Li analisaram as pesquisas efectuadas no sistema de pesquisas de imagens na Web WebSeek [Smith98] e concluíram que os utilizadores utilizam especificações de

alto nível semântico. Outro estudo relevante [Spink00] examina as interrogações efectuadas pelos utilizadores no motor de busca Excite, filtrando apenas as que incluíam pesquisas multimédia (imagens, áudio e vídeo). Este estudo permitiu concluir que os utilizadores pesquisam informação multimédia na Web de uma forma idêntica quando procuram informação textual, acrescentando apenas um atributo multimédia (por exemplo: *music*, *movies*, jpeg) e que necessitam de 3 termos em média por pesquisa. Jansen e outros autores [Jansen00] verificaram que os utilizadores ao efectuarem pesquisas textuais utilizam tipicamente 2 termos por pesquisa. Este estudo permitiu ainda concluir que: o intervalo de tempo de uma sessão típica é de 10 a 15 minutos; são feitas, em média, duas pesquisas por sessão; os utilizadores não utilizam as funcionalidades avançadas da pesquisa e que apenas visualizam os dez primeiros resultados de cada pesquisa.

Apesar da inegável utilidade destes estudos, na medida em que fornecem informação sobre as características dos utilizadores e das pesquisas de imagens na Web é fundamental efectuar uma análise mais detalhada das características das interfaces de interrogação e de apresentação de resultados, determinar o paradigma de especificação mais adequado para cada tipo de utilizador (por exemplo: textual ou visual) e analisar o comportamento dos utilizadores na utilização destes sistemas. Este estudo é importante não só para a compreensão das necessidades dos utilizadores, mas também para facilitar o *design* de interfaces de sistemas de recuperação de imagens [Choi03; Cunningham04].

Este artigo apresenta um estudo de usabilidade aos utilizadores de sistemas de pesquisa de imagens na Web como o objectivo de compreender qual o método mais adequado para a especificação das suas necessidades. Para tal, são analisados alguns aspectos, tais como: percepção do modo como os utilizadores formulam as interrogações e o que esperam do sistema; identificação dos casos nos quais é preferível uma pesquisa baseada no conteúdo ou uma pesquisa baseada em texto; análise da correlação entre perfis de utilizadores e tipo de especificação de pesquisa; identificação do número de termos por pesquisa textual e respectivas características; identificação do número de imagens por pesquisa caso a especificação seja visual; grau de satisfação acerca das imagens retornadas e análise da interface de resultados da pesquisa.

O artigo está organizado da seguinte forma: a secção 2 apresenta os diferentes paradigmas de especificação de pesquisas de imagens na Web e os sistemas mais representativos para cada abordagem; a secção 3 descreve o plano de avaliação efectuado; na secção 4 é realizada uma análise comparativa dos resultados obtidos e as principais implicações no *design* de interfaces para sistemas de pesquisa de imagens na Web; na secção 5 é apresentado um protótipo de uma interface de pesquisa de imagens na Web e por último, a secção 6, resume as

principais contribuições do trabalho efectuado e discute os aspectos de continuidade do tema tratado.

2. PESQUISA DE IMAGENS NA WEB

Esta secção apresenta os diversos paradigmas de especificação das pesquisas de imagens na Web, com o objectivo de enquadrar o estudo de usabilidade realizado (secção 3). Adicionalmente, apresentam-se os sistemas mais representativos, para cada paradigma.

A especificação de pesquisas de imagens no contexto da Web pode ser efectuada através de três paradigmas principais, sem considerar as possíveis combinações: especificação textual da pesquisa; especificação visual da pesquisa e pesquisa ou navegação por directório de imagens. Na pesquisa visual podem ainda distinguir-se as pesquisas por imagem de exemplo, por ícone e por esboço. A Tabela 1 apresenta um resumo de cada um destes paradigmas a nível da especificação de pesquisas e dos resultados apresentados aos utilizadores.

2.1 Especificação textual da pesquisa

Em muitos casos descrever uma imagem, de um modo textual, é muito mais fácil do que procurar ou desenhar uma imagem de exemplo. Apesar das imagens e as palavras fornecerem significados distintos, os utilizadores preferem as palavras-chave para efectuar a interrogação e só depois, de modo visual, confirmar a relevância de cada resultado obtido [Ding99].

Os sistemas de pesquisa de imagens na Web recorrem tipicamente ao texto agregado à imagem (por exemplo: *caption* da imagem, texto alternativo, título da página Web e nome do ficheiro) para encontrar as imagens mais relevantes para a especificação do utilizador. O utilizador introduz um conjunto de palavras-chave ou uma descrição da imagem pretendida e são utilizadas técnicas convencionais de pesquisa textual sobre essas anotações.

Apesar deste método ser o mais intuitivo, na medida em que é utilizado o vocabulário do utilizador, não está isento de problemas. Os sistemas de pesquisa de imagens baseados em texto assumem que existem descrições textuais das imagens. Contudo, os utilizadores são relutantes em anotar informação visual na Web sendo frequente não existir nenhuma descrição textual. Por outro lado, as anotações produzidas para cada imagem são naturalmente subjectivas e dependentes dos criadores da página Web. Por último, a utilização de anotações textuais por si podem revelar-se insuficientes, especialmente para utilizadores interessados nas características visuais de uma imagem, nos objectos que nela estão representados e no assunto sobre o qual versam (por exemplo: uma imagem com um cálice de vinho pode referir-se a uma missa cristã). Consequentemente, a eficiência dos motores de busca de imagens na Web baseados apenas neste tipo de pesquisa pode apresentar algumas limitações.

A maior parte dos mais conhecidos motores de busca de imagens na Web inclui já interfaces de pesquisa de imagens que utilizam este paradigma, como é o caso do

Google Images¹, o Flickr², o ImageRover [Taycher97], Diogenes [Aslandogan00] e o WebSEEk [Munson01].

	Especificação	Resultado
Textual	Descrição textual ou palavra chave sobre a imagem pretendida	São retornadas as imagens que contêm as anotações textuais mais relevantes com as utilizadas na pesquisa
Visual	Imagem de Exemplo: o utilizador selecciona ou fornece uma imagem com características semelhantes à imagem pretendida	São devolvidas imagens que contêm cor, textura ou formas semelhantes à imagem fornecida
	Esboço: o utilizador desenha um esboço da imagem pretendida	O esboço é comparado com as imagens da base de dados e são retornadas as mais semelhantes
	Ícones Semânticos: o utilizador coloca ícones representativos no local onde espera que apareçam na imagem	Retorna imagens que contêm as características visuais no local especificado pelo utilizador
Directório	O utilizador navega através de uma árvore ou grafo de temas relacionados com a imagem pretendida	São retornadas as imagens que estão classificadas nas categorias escolhidas pelo utilizador

Tabela 1 – Paradigmas de especificação de pesquisas de imagens.

2.2 Especificação visual da pesquisa

A pesquisa de imagens baseada no conteúdo assenta na caracterização de propriedades primitivas, tais como: a cor, a forma e textura que podem ser automaticamente extraídas das próprias imagens. Este paradigma revelou-se essencial para complementar as anotações textuais sobre as imagens de modo a produzir resultados mais relevantes para o utilizador [Kherfi04b].

Nos sistemas de pesquisa de imagens baseada no conteúdo, a formulação da interrogação pode apresentar uma dificuldade acrescida e têm sido propostas diferentes abordagens, nomeadamente, os paradigmas de pesquisa por:

- imagem de exemplo;
- esboço;
- ícones.

Os primeiros sistemas de CBIR (*Content Based Image Retrieval*) disponibilizam interfaces nas quais o utilizador especifica as características pretendidas a nível da cor, textura e forma através da escolha de uma imagem exemplo (*query-by-similar-images*) [Gevers00]. Alguns exemplos de sistemas que utilizam este tipo de paradigma de pesquisa são: QBIC [Flickner95], WebSeek [Smith97], PicToSeek [Gevers97] e ImageRover [Taycher97].

Contudo este paradigma apresenta algumas limitações. O problema de determinar a melhor imagem ou conjunto de imagens inicial é denominado por *page zero problem*. Para resolver este problema o sistema deverá fornecer ao utilizador um conjunto de imagens candidatas que sejam representativas de todo o conteúdo visual da Web. Outra solução utilizada pelos sistemas WebSeek e o ImageRover, consiste na especificação de uma pesquisa textual inicial, após a qual é possível realizar um refinamento textual ou visual. O sistema PicToSeek requer que o utilizador forneça inicialmente uma imagem exemplo. Outra aproximação, para atenuar este problema, consiste na utilização de exemplos positivos juntamente com exemplos negativos. Este processo é denominado de retorno de relevância [Kherfi04a]. O retorno de relevância consiste em ajustar automaticamente uma pesquisa, usando informação fornecida pelo utilizador sobre a relevância dos objectos previamente obtidos. Os utilizadores podem assim, refinar iterativamente a pesquisa, através da selecção dos exemplos mais relevantes, bem como dos irrelevantes. Exemplos de sistemas que utilizam este paradigma são: o PicToSeek [Gevers97], o Fire [Deselaers04], o ImageRover [Taycher97], o RIME [Chang98] e o Retrievr³.

No paradigma de especificação por esboço, os utilizadores podem submeter um esboço, clicar numa paleta de texturas ou simplesmente seleccionar uma forma que lhes interesse. O sistema identifica as imagens armazenadas que têm o maior nível de similaridade com o esboço [Goodrum00]. Este tipo de especificação permite ao utilizador especificar quais as partes da imagem que são mais importantes. A um nível de abstracção mais elevado, este tipo de pesquisas pode também referenciar os objectos retratados, eventos ou mesmo emoções presentes na imagem [Zhang05].

Contudo, a pesquisa por esboço apresenta alguns problemas. Não é evidente qual a importância que deverá ser assumida quanto à disposição espacial dos objectos no esboço efectuado pelo utilizador. Por exemplo, se o utilizador desenhar um círculo na parte inferior do esboço, a sua verdadeira necessidade de informação é de círculos na parte inferior da imagem ou apenas círculos independentemente da sua disposição espacial. Existe ainda, uma dificuldade na especificação da importância do fundo do desenho (*background*) relativamente à parte da frente (*foreground*). Por exemplo, se o utilizador desenhar uma bola e existir uma imagem de uma bola contra um fundo branco e outra com um miúdo com uma bola, qual será a mais relevante para o utilizador. Não existe uma resposta correcta às perguntas anteriores, uma vez que cada utilizador tem as suas (diferentes) preferências. São exemplos de sistemas que utilizam este tipo de pesquisa, o Retrievr³, o ImageScape [Lew00] e o QBIC [Flickner95], nos quais o utilizador tem à disposição ferramentas simples de desenho.

¹ <http://images.google.com/>

² <http://www.flickr.com/>

³ <http://labs.systemone.at/retrievr/>

No caso do ImageScope, o utilizador tem ainda à disposição um conjunto de ícones semânticos que se referem a diferentes conceitos (por exemplo: o mar, árvores, pessoas) que o utilizador pode colocar na pesquisa, indicando desde modo onde é que deverá aparecer na imagem [Lew97].

2.3 Directório de Imagens

No caso da pesquisa de imagens através de um directório de imagens, o motor de busca oferece uma taxionomia baseada em temas tais como: “arte e cultura”, “desporto”, “regiões do mundo”. Cada tema pode ser por sua vez dividido em sub-temas criando uma hierarquia de categorias (por exemplo: o tema “regiões do mundo” pode ser dividido nos sub-temas, “Europa”, “África” e “América”). Podem ainda existir *links* entre os diferentes temas, constituindo um dicionário, que permite ao utilizador navegar entre um grande conjunto de temas, cada um com um conjunto de imagens associado.

Um exemplo deste tipo de sistemas é o WebSEEk, no qual a navegação através do directório é associada à pesquisa textual realizada sobre um determinado tema [Munson01].

3. PLANO DE AVALIAÇÃO

Apesar do grande número de possíveis sistemas de pesquisa de imagens na Web, não existem ainda muitos estudos efectuados acerca da usabilidade destes sistemas. O trabalho aqui apresentado tem, portanto, como objectivo geral estudar as interacções dos utilizadores na utilização destes sistemas, através de uma análise detalhada das interrogações efectuadas pelos utilizadores e da observação do seu comportamento quando utilizam estes sistemas.

O presente estudo tem os seguintes objectivos específicos:

- Agrupar os utilizadores de acordo com os seus perfis e identificar as preferências de cada categoria;
- Perceber o modo como os utilizadores formulam as interrogações e o que esperam do sistema;
- Identificar os casos nos quais é preferível uma pesquisa visual ou uma pesquisa baseada em texto;
- No caso das pesquisas textuais, identificar o número de termos por pesquisa e respectivas características;
- No caso das pesquisas visuais, analisar o número de imagens por pesquisa;
- Obter retorno sobre o grau de satisfação acerca das imagens retornadas;
- Analisar a relação entre as preferências dos utilizadores e os seus perfis.

A extracção deste conjunto de características visa compreender o método mais adequado para a especificação das imagens pretendidas pelos utilizadores.

Para a elaboração do plano de avaliação, foram escolhidos os sistemas mais representativos das abordagens

referidas na secção anterior. Apesar de ter sido feita uma análise documental mais exaustiva sobre outros sistemas de pesquisa de imagens na Web, o facto de estes não estarem disponíveis na Web, impediu a sua utilização no plano de avaliação. Assim, como sistema mais representativo das pesquisas de imagens com especificação textual foi escolhido o Google Images que é, simultaneamente, um motor de busca já conhecido da maioria das pessoas. O sistema WebSEEk permite, simultaneamente, pesquisas com especificação textual, pesquisas através de um directório e testar a utilização de uma ferramenta de retorno de relevância. Por último, como sistema representativo da especificação de pesquisas por esboço foi escolhido o motor de busca Retrievr que permite ainda a realização de pesquisas através de imagens de exemplo.

Com o objectivo de avaliar estes três sistemas, foi concebido um teste de usabilidade composto por três partes distintas: pré e pós questionário e uma entrevista com guião. O pré questionário é constituído por um conjunto de perguntas que visam obter informações sobre o perfil dos utilizadores e sobre a sua experiência de pesquisa de imagens na Web.

A entrevista com guião tem como objectivo a realização de um conjunto de tarefas. A tarefa inicial consiste na exploração livre do sistema por parte do utilizador com o objectivo de obter uma opinião global para cada sistema a nível de dificuldades de aprendizagem e de especificação da pesquisa. A segunda tarefa consiste na realização de uma pesquisa específica: “procurar imagens de rosas vermelhas”. Com esta tarefa pretende-se obter a opinião dos utilizadores sobre o ecrã de resultados de cada sistema. A última tarefa consiste na pesquisa de imagens semelhantes a três imagens fornecidas inicialmente. Estas imagens foram escolhidas utilizando um critério que permitisse verificar a existência de diferentes níveis de dificuldade no desenho do esboço, ou especificação textual (Figura 1).



Figura 1: Imagens utilizadas no plano de avaliação.

Uma vez que apenas é fornecida uma imagem e não um nome ou descrição verbal do que se pretende, é possível, através deste teste, perceber de que modo os utilizadores especificam a pesquisa que pretendem efectuar. No teste ao sistema WebSEEk, são ainda efectuadas pesquisas utilizando o directório existente, de modo a perceber a opinião dos utilizadores sobre este tipo de pesquisa. É ainda analisada a utilização do retorno de relevância, com o objectivo de perceber se esta é ou não uma ferramenta útil. No teste ao sistema Retrievr, além dos testes anteriormente mencionados, efectuados recorrendo ao esboço, são também efectuadas pesquisas por imagem de exemplo, recorrendo a ficheiros fornecidos.

O pós questionário (ou de satisfação) foi construído com base no QUIS⁴ (*Questionnaire for User Interaction Satisfaction*) e no questionário normalizado da Digital Equipment Corporation, o SUS⁵ (*System Usability Scale*). Este questionário engloba um conjunto de perguntas, nas quais os utilizadores fazem uma comparação, sobre diversos aspectos dos diferentes sistemas, classificando cada um deles numa escala de 1 (Muito Mau) a 6 (Muito Bom). É ainda pedido aos utilizadores que indiquem quais os sistemas que utilizariam para realizar as tarefas no dia a dia e também que façam um comentário final sobre cada sistema.

Para uma correcta avaliação dos sistemas, e baseado no estudo de Nielsen [Nielsen00], o plano de avaliação foi realizado a 10 pessoas distintas, com perfis diversificados, que abrangem 3 pessoas da área de gestão, 2 de ciências, 1 de artes e 4 de informática; bem como diferentes classes etárias (“18-24”, “25-34”, “35-45”, “>45”) e diferentes níveis de experiência. Durante a realização do plano de avaliação todos os utilizadores testaram os três sistemas de pesquisa de imagens.

4. ANÁLISE DOS RESULTADOS

Esta secção apresenta os principais resultados obtidos para cada sistema através do pré questionário e das entrevistas com guião, e uma comparação entre os diferentes sistemas através do pós questionário.

4.1 Google Images

O motor de busca Google Images (Figura 2) obteve em geral boas classificações, sendo caracterizado como muito fácil de utilizar e também muito fácil de aprender a utilizar; é ainda classificado como tendo uma boa aparência e uma boa organização do ecrã de pesquisa.

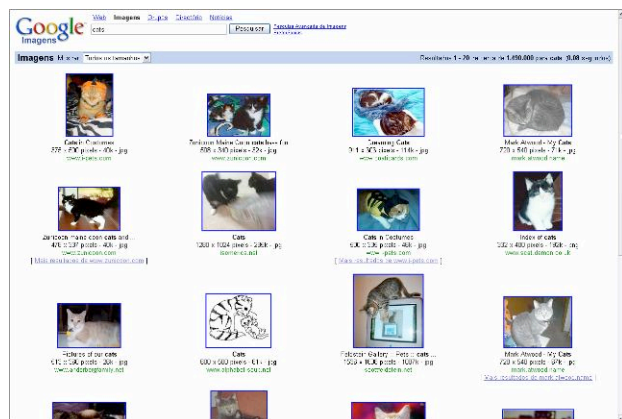


Figura 2: Ecrã de resultados do Google Images.

Contudo, alguns utilizadores referiram a necessidade de existirem menos imagens por página de resultados, cada imagem apresentar uma maior dimensão e eliminar o *scroll* nas páginas de resultados. Por outro lado, é ainda referido por um número considerável de utilizadores que após a obtenção da página de resultados, o número de

⁴ www.lap.umd.edu/quis/index.html

⁵ www.usability.serco.com/trump/documents/suschart.doc

passos até obter a imagem escolhida é excessivo. A recomendação geral é a de que, o utilizador após a escolha da imagem, a visualize logo no formato real, e só depois, caso pretendido, seja redireccionado para o site onde se localiza a imagem. A principal dificuldade expressada pelos utilizadores é encontrar os termos adequados para especificar a sua necessidade de informação.

Em média, os utilizadores fazem duas pesquisas para encontrar a imagem pretendida e não visitam mais do que a primeira página de resultados. O tempo médio para encontrar as imagens pretendidas é de 29 segundos.

4.2 WebSEEk

O sistema WebSEEk permite, simultaneamente, pesquisas com especificação textual e pesquisas através de um directório e a utilização de uma ferramenta de retorno de relevância (Figura 3). O motor de busca WebSEEk obteve, ao contrário dos outros sistemas, uma má classificação, tanto ao nível da aprendizagem e facilidades de utilização, como da aparência e da organização do ecrã de pesquisa.

Uma das características exploradas neste sistema, é a existência de um directório para efectuar as pesquisas. No entanto, este tipo de pesquisa ou navegação obteve uma opinião bastante negativa por parte dos utilizadores, nomeadamente no que diz respeito à organização dos temas. O número de passos necessário para encontrar as imagens através da utilização da navegação no directório é também considerado excessivo. Deste modo, apenas 20% dos inquiridos afirmam preferir utilizar uma pesquisa com navegação no directório em vez de uma pesquisa textual.

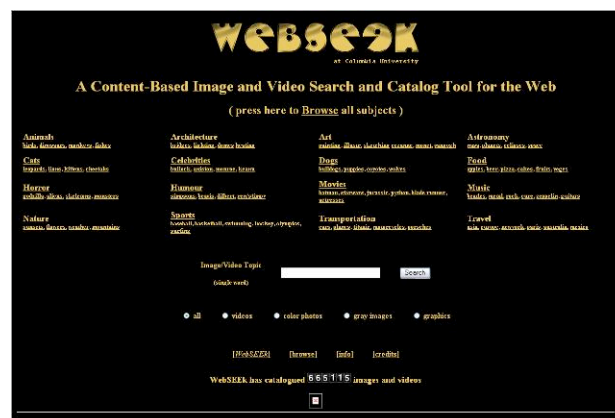


Figura 3: Ecrã de pesquisa do WebSEEk.

A outra característica que distingue este sistema dos restantes é a possibilidade de utilização de uma ferramenta de retorno de relevância. Esta ferramenta, apesar de não ser considerada muito intuitiva, é considerada como tendo alguma utilidade.

Neste sistema são várias as dificuldades sentidas pelos utilizadores. A mais notória, expressa pela totalidade dos inquiridos, é a impossibilidade de utilizar mais do que um termo na pesquisa textual. Para além do utilizador não se aperceber deste aspecto porque não existe qual-

quer informação sobre o número limite de termos, torna à partida a especificação da pesquisa mais limitada e vasta.

Tal como no sistema Google Images, alguns inquiridos consideram existirem passos desnecessários até à obtenção da imagem. Os utilizadores sugeriram que após ter sido submetida a pesquisa, deverão aparecer logo as imagens resultantes e não apenas um *link* para a página com os resultados. Um factor que influenciou negativamente o teste é a existência de erros. Foram frequentes as situações em os utilizadores não conseguiram aceder às imagens retornadas pelo sistema na página de resultados.

Em média, os utilizadores fazem duas pesquisas e visitam 2,5 páginas de resultados. O tempo médio para encontrar as imagens pretendidas é de 1 minuto e 26 segundos.

4.3 Retrievr (Esboço)

No caso das pesquisas através de esboço no motor de busca Retrievr, as classificações são em geral boas, obtendo apenas uma classificação mais baixa na facilidade de utilização. Este resultado coincide com a principal dificuldade referida pelos utilizadores na especificação da pesquisa através do desenho de um esboço, especialmente quando realizado através de uma ferramenta de desenho tradicional.

No que diz respeito ao desenho do esboço, apesar de o número de cores disponível ser considerado adequado, os utilizadores referem que as opções disponíveis, que consistem apenas num conjunto de círculos de diferentes tamanhos, não são suficientes. Como tal, são dadas diversas sugestões de ferramentas que devem ser incluídas, tais como, formas geométricas, traços mais finos, uma borracha e a inclusão de texturas predefinidas. Uma característica apontada como negativa por diversos utilizadores, é a inexistência de um botão que permita submeter a pesquisa, uma vez que o sistema submete automaticamente o esboço de cada vez que se realiza uma alteração ao desenho anterior.

Outra dificuldade que foi referenciada é a impossibilidade de obter bons resultados em tempo útil. De facto, em média são necessários 2 minutos e 46 segundos para os utilizadores encontrarem as imagens pretendidas e são necessárias em média 3 pesquisas para encontrar as imagens relevantes

4.4 Retrievr (Imagem Exemplo)

No teste ao sistema Retrievr por pesquisa de imagem exemplo (Figura 4), os utilizadores classificaram esta funcionalidade como sendo útil e fácil de utilizar. No entanto, a maioria dos utilizadores referiram que os resultados obtidos são maioritariamente irrelevantes, e que para que se possa efectuar este tipo de pesquisas, é necessário ter já disponível, à partida, um grande conjunto de imagens, ou uma imagem semelhante.

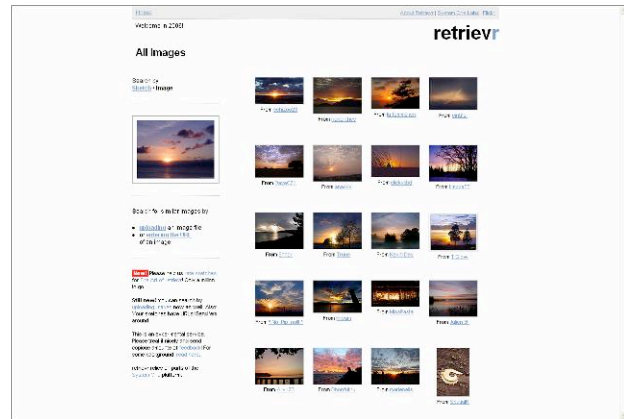


Figura 4: Pesquisa por imagem de exemplo no Retrievr.

4.5 Comparação e Conclusões

Como comparação entre os diferentes sistemas, é importante notar que o tempo necessário para encontrar as imagens pretendidas é muito menor no Google Images (29 segundos) do que nos restantes sistemas. No caso do WebSEEK (1 minuto e 26 segundos) devido essencialmente à dificuldade em encontrar imagens relevantes, e no caso do Retrievr, devido não só à dificuldade para encontrar as imagens, mas também devido ao tempo que demora a desenhar cada esboço.

A Tabela 2 apresenta a mediana dos resultados obtidos pelos diversos sistemas analisados numa escala entre 1 e 6. Tal como foi já referido, o Google Images obtém praticamente sempre a melhor classificação. No extremo oposto está o WebSEEK, que obtém as piores classificações.

	Google	Webseek	Retrievr (esboço)	Retrievr (img ex)
Aprendizagem	6,0	3,0	5,0	5,0
Tempo de Aprendizagem	5,5	3,5	5,0	6,0
Facilidade de Utilização	6,0	3,0	4,0	5,0
Facilidade de Especificação	6,0	2,5	3,5	5,5
Rapidez de Especificação	6,0	3,0	3,0	5,0
Ajuda Fornecida	5,0	3,0	4,5	4,5
Nº de Passos	6,0	2,5	3,5	5,0
Sequência de Acções	5,5	3,0	4,5	4,5
Voltar Atrás	6,0	4,5	4,5	5,0
Funcionalidades	5,0	2,5	5,0	5,0
Simplicidade	5,5	3,5	4,5	5,0
Aparência	5,0	3,0	5,0	5,0
Qualidade das Imagens	5,0	3,0	5,0	5,0
Eficiência	5,5	3,0	3,0	3,5
Nº de Imagens Retornadas	5,0	4,0	5,0	5,0
Utilização do Sistema	6,0	2,5	3,5	4,0

Tabela 2: Classificações comparativas entre os diferentes sistemas.

Após ter sido feita a análise inicial dos dados, procedeu-se ainda a um cruzamento de dados de forma a analisar se existem diferenças significativas nas respostas dos diferentes perfis de utilizadores. Contudo, em geral as respostas são muito homogêneas não se notando a diferença entre os diferentes perfis dos utilizadores.

As respostas obtidas no sistema Google Images são idênticas para os diferentes perfis. Apenas na avaliação dos restantes sistemas se verificou alguma diferenciação de respostas, especificamente entre os diferentes escalões etários e também entre os inquiridos das áreas de gestão e administração e os das restantes áreas.

Os inquiridos pertencentes ao escalão etário “idade superior a 45” e das áreas de gestão e administração, comparativamente com os restantes, classificam sempre pior o WebSEEK e o Retrievr. Tal facto não foi observado na avaliação do sistema Google Images. Verificou-se também que os inquiridos com este perfil apresentam mais dificuldades de utilização destes sistemas, demoram mais tempo a realizar uma pesquisa e que tipicamente visualizam mais páginas de resultados.

Independentemente do perfil de cada inquirido a resposta à pergunta, “Qual o sistema que utilizariam no dia a dia”, foi unânime em considerar o sistema Google Images, excepto um inquirido que prefere o sistema WebSEEK. A maioria dos inquiridos refere que o sistema que recomendavam é o Google Images complementado com pesquisa por imagens de exemplo, por esboço e com retorno de relevância.

Relativamente à experiência em pesquisa de imagens, 3 pessoas efectuam várias vezes por semana este tipo de pesquisas, 1 utiliza uma vez por semana e 6 apenas ocasionalmente. Todas as pessoas pesquisam imagens por motivos de entretenimento e 7 também por motivos profissionais. Por outro lado, já todos tinham conhecimento do sistema Google Images e duas pessoas conheciam também o sistema Retrievr. Apesar destes diferentes níveis de experiência, os resultados dos testes foram, uma vez mais, homogêneos.

Como conclusão, é possível observar que os utilizadores mais novos classificam melhor os novos paradigmas de especificação de pesquisas. No entanto, devido à homogeneidade das restantes respostas obtidas, é também possível concluir que os diferentes utilizadores têm preferências muito semelhantes, não sendo por isso possível estabelecer grandes diferenciações quanto aos perfis e respectivas características.

No caso das pesquisas textuais, é possível identificar que o número de termos por pesquisa é, em média, de 2 termos, o que confirma o valor apresentado em [Spink00] e [Jansen00] nos quais se refere que o valor médio é de dois termos por pesquisa. Fica também confirmado que os utilizadores, em geral, vêm apenas a primeira página de resultados. Por outro lado, é também possível observar que os utilizadores não utilizam as funções avançadas de pesquisa e utilizam quase sempre palavras-chave

(por exemplo: “montanha neve”) em detrimento da linguagem natural (ex.: “montanhas com neve”). Neste tipo de pesquisas, os utilizadores estão, em geral, satisfeitos com as imagens retornadas.

No caso das pesquisas visuais, os utilizadores demoram muito tempo para obter resultados relevantes, e em média é necessário desenhar cerca de três imagens para encontrar os resultados pretendidos. O número de pesquisas efectuadas e o tempo necessário para encontrar as imagens estão directamente relacionados com a ainda dificuldade deste tipo de sistemas em encontrar imagens relevantes.

Apesar do reconhecimento por parte dos utilizadores que em algumas situações a pesquisa textual não é suficiente para satisfazer a sua necessidade de informação, os restantes paradigmas ainda não oferecem uma alternativa eficaz e, como tal, são descartados pelos utilizadores. Devem no entanto ser fornecidos como complemento à pesquisa textual, sendo apenas utilizados quando o primeiro paradigma falha em termos de imagens relevantes.

5. PROTÓTIPO

Com vista a dar continuidade ao estudo efectuado, foi construído um protótipo de uma interface que tenta apresentar uma possível resolução de algumas das dificuldades apresentadas pelos utilizadores. Este protótipo utiliza, como abordagem primária, uma especificação textual da pesquisa (Figura 5) e é complementado com a especificação de pesquisas através de imagens exemplo (Figura 6) ou através de um esboço (Figura 7).

Uma vez que os utilizadores estão globalmente satisfeitos com a especificação textual das pesquisas no sistema Google Images, a interface construída no protótipo na abordagem textual é semelhante.



Figura 5: Interface de pesquisa textual e Interface de resultados da pesquisa.

No caso das pesquisas através de uma imagem exemplo, a implementação é também baseada na interface utilizada no sistema Retrievr com algumas diferenças na organização dos resultados obtidos.



Figura 6: Interface de pesquisa por imagem de exemplo.

A pesquisa por esboço é, no entanto, bastante diferente da utilizada no Retrieve. As alterações efectuadas respeitaram as sugestões efectuadas pelos utilizadores na inclusão de mais opções de desenho. Como tal, foram adicionadas várias opções para o desenho do esboço, tal como se pode observar na Figura 7.

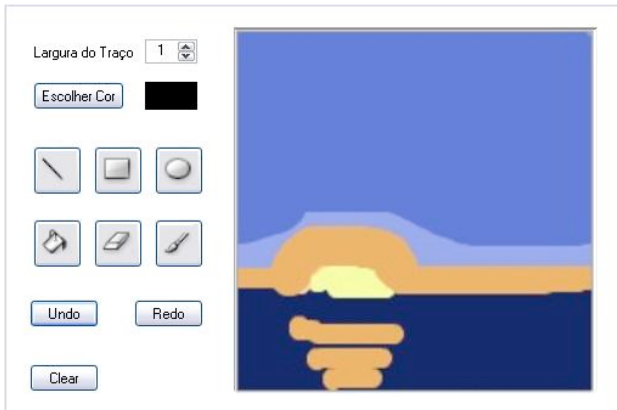


Figura 7: Formulário para desenhar o esboço a pesquisar.

Na interface da página de resultados foi introduzido o conjunto de alterações mais significativo sugerido pelos utilizadores. Para evitar passos desnecessários para a obtenção de uma imagem em tamanho real, quando se selecciona a imagem miniatura obtém-se imediatamente a imagem original, evitando o passo intermédio. Caso o utilizador seleccione o URL do site que se encontra por baixo da imagem (ver Figura 8), então o utilizador é redireccionado para o site onde a imagem se localiza.

Outra alteração efectuada consistiu na redução do número de imagens retornadas, de modo a evitar o *scroll* das páginas de resultados. Foi também adicionado um pequeno ícone de uma lupa a cada imagem retornada (ver Figura 8), de modo a permitir que após o utilizador ter efectuado uma pesquisa textual, possa utilizar uma das imagens para efectuar uma pesquisa por imagem de exemplo.



Figura 8: Uma pequena lupa ao lado de cada imagem retornada permite utilizá-la como imagem de exemplo.

6. CONCLUSÕES E TRABALHO FUTURO

Neste artigo apresentou-se um estudo de usabilidade que envolveu a análise das interrogações e a observação dos utilizadores na utilização de sistemas de pesquisa de imagens na Web. O plano de avaliação proposto envolveu a elaboração de pré e pós questionários e entrevistas com guião a 10 utilizadores com diferentes perfis de utilização. Apesar da amostra de utilizadores não ser estatisticamente significativa, o estudo realizado permitiu detectar as principais dificuldades dos utilizadores destes sistemas. A análise dos resultados revelou que os utilizadores preferem efectuar pesquisas textuais e utilizam em média 2 termos por pesquisa. O recurso a outros métodos de especificação da pesquisa (por esboço ou imagem exemplo) apenas é utilizado em caso de insucesso do método textual. O resultado do estudo permitiu ainda extrair as principais implicações no *design* de interfaces para sistemas de pesquisa de imagens na Web. Com base nestes indicadores foi realizado um protótipo da interface de pesquisa e da interface da página de resultados com o objectivo de apresentar uma possível resolução de algumas das dificuldades levantadas pelos utilizadores.

Uma vez que os resultados da avaliação de um sistema estão, em grande parte, condicionados pela sua capacidade de retornar ao utilizador as imagens que este pretende, é também importante que seja feita uma avaliação rigorosa da eficiência e eficácia dos motores de busca de imagens na Web baseados em pesquisa por conteúdo, de modo a permitir melhorar a qualidade e relevância dos resultados obtidos. Por outro lado, a utilização do protótipo desenvolvido num futuro teste de usabilidade poderá permitir um teste mais eficiente, no qual os diferentes paradigmas de pesquisa utilizam a mesma interface.

7. REFERÊNCIAS

- [Aslandogan00] Y. Aslandogan, C. Yu. Evaluating Strategies and Systems for Content Based Indexing of Person Images on the Web. *Proceedings of the ACM Multimedia*, 2000.
- [Chang98] E. Chang et. al. RIME: A Replicated Image Detector for the World-Wide Web. *Proceedings of the SPIE Symposium of Voice, Video and Data Communications*, pp. 58-67, 1998.

- [Choi03] Y. Choi, E. Rasmussen. Searching for Images: The Analysis of Users Queries for Image Retrieval in American History. *Journal of the American Society for Information Science and Technology*, pp. 498-511, 2003.
- [Cunningham04] S. Cunningham et. al. How People Describe Their Image Information Needs: A Grounded Theory Analysis of Visual Arts Queries. *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pp. 47-48, 2004.
- [Deselaers04] T. Deselaers et. al. FIRE - Flexible Image Retrieval Engine: ImageCLEF 2004 Evaluation. *Proceedings of the CLEF 2004 Workshop*, pp. 535-544, 2004.
- [Ding99] W. Ding. Cognitive Processing of Multimodal Surrogates for Video Browsing. *Unpublished Doctoral Dissertation, School of Library and Information Services, University of Maryland*, 1999.
- [Flickner95] M. Flickner et al. Query by Image and Video Content: The QBIC System. *IEEE Computer*, Volume 28, Nº 9, pp. 23-32, 1995.
- [Gevers97] T. Gevers and A. Smeulders. PicToSeek: A Content-Based Image Search System for the World Wide Web. *Proceedings of the Visual97 Conference*, pp. 93-100, 1997.
- [Gevers00] T. Gevers et. al. Classification of Images on Internet by Visual and Textual Information. *Proceedings of the Internet Imaging*, 2000.
- [Goodrum00] A. Goodrum. Image Information Retrieval: An Overview of Current Research. *Informing Science*, Volume 3, Nº 2, pp. 63-66, 2000.
- [Gudivada95] V. Gudivada, V. Raghavan. Content Based Image Retrieval Systems. *IEEE Computer*, Volume 28, Nº 9, pp. 18-22, 1995.
- [Hastings95] S. K. Hastings. Query Categories in a Study of Intellectual to Digitized Art Images. *Proceedings of the 58th Annual Meeting of the American Society for Information Science*, Volume 32, pp. 3-8, 1995
- [Holt94] B. Holt, L. Hartwick. Retrieving Art Images by Content: The UC Davis QBIC Project. *Aslib Proceedings*, Volume 46, Nº 10, pp. 243-248, 1994.
- [Jansen00] B. Jansen et. al. Searching for Multimedia: An analysis of audio, video, and image web queries. *World Wide Web Journal*, Volume 3, Nº 4, pp. 249-254, 2000.
- [Keister94] L. A. Keister. User Types and Queries: Impact on Image Access Systems. *R. Fidel, T.B. Hahn, E.M. Rasmussen, & P.J. Smith (Eds.), Challenges in Indexing Electronic Text and Images*. Medford, NJ: *Learned Information for the American Society for Information Science*, pp. 7-22, 1994.
- [Kherfi04a] M.L. Kherfi, D. Ziou: Image retrieval based on feature weighting and relevance feedback. *Proceedings of the 2004 International Conference on Image Processing (ICIP 2004)*, pp. 689-692, 2004.
- [Kherfi04b] M. L. Kherfi, et. al. Image Retrieval From the World Wide Web: Issues, Techniques, and Systems. *ACM Computing Surveys*, Volume 36, Nº 1, pp. 35-67, 2004.
- [Lawrence99] S. Lawrence, C. Giles. Text and Image Metasearch on the Web. *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 829-835, 1999.
- [Lew00] Michael S. Lew. Next-Generation Web Searches for Visual Content. *IEEE Computer*, Volume 33, Nº 11, pp. 46-53, 2000
- [Lew97] M. Lew, et. al. Webcrawling Using Sketches. *Proceedings of the Visual97 Conference*, pp. 77-84, 1997.
- [Munson01] E. Munson. To Search for Images on the Web, Look at the Text, Then Look at the Images. *Proceedings of the First International Workshop on Web Document Analysis*, pp. 39-42, 2001.
- [Nielsen00] J. Nielsen, Why You Only Need To Test With 5 Users, 2000.
<http://www.useit.com/alertbox/2000319.htm>.
- [Smith97] J. R. Smith and S. F. Chang, Visually Searching the Web for Content. *IEEE Multimedia*, pp. 12-20, 1997.
- [Spink00] A. Spink et. al. Multimedia Web Queries: Implications for Design. *International Conference on Information Technology: Coding and Computing (ITCC '01)*, 2001.
- [Taycher97] L. Taycher et. al. Image Digestion and Relevance Feedback in the ImageRover WWW Search Engine. *Proceedings of the 2nd International Conference on Visual Information*, pp. 85-94, 1997.
- [Turner95] J. M. Turner. Comparing User-Assigned Terms with Indexer-Assigned Terms for Storage and Retrieval of Moving Images: Research Results. *Proceedings of the 58th Annual Meeting of the American Society for Information Science*, Volume 32, pp. 9-12, 1995.
- [Zhang05] C. Zhang et. al. User Term Feedback in Interactive Text-Based Image Retrieval. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 51-58, 2005.

Above and Beyond: Outer Space as a Metaphor for Document Browsing and Visualization

Daniel P. Gonçalves Luis Bilo Daniel Gonçalves Manuel J. Fonseca Joaquim A. Jorge
Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
daniel.goncalves@inesc-id.pt, mjf@inesc-id.pt, jorgej@acm.org

Abstract

Currently, any ordinary user has hundreds or even thousands of electronic documents stored on his computer. Usually, those documents are archived using an hierarchically-organized file system, making their retrieval at a later time a difficult task. In this paper we present a novel approach for document browsing and visualization based on the Outer Space metaphor. While users "travel" from galaxies to planets and their satellites, the search criteria become narrower eventually allowing users to find their documents. With our approach, users can find a specific file in no more than four clicks. We developed a preliminary prototype, using a user centered approach, through the development of low-fidelity prototypes and heuristic evaluations.

Keywords

User Interfaces, Heuristic Evaluation, Information Visualization

1. Introduction

It is nowadays common for users to handle hundreds of electronic documents. Unfortunately, the way in which those documents are organized makes this a cumbersome task. Indeed, documents are nothing more than files in a hierarchically-organized file system. When trying to retrieve a file, users can resort to little more information than the file's location in the hierarchy. However, such a classification is fraught with problems. When storing a document, where to place it is often not a trivial decision. More than one place in the hierarchy (or no place at all!) might seem adequate. Also, what seems a good classification at one point in time might not be the one remembered at a later time. As such, retrieving the document can become impossible.

The aforementioned difficulties result from the fact that while the users are handling documents, the computer handles files. A document is something a user remembers fairly well, as it was read or written for a reason, can have memorable contents, and was handled in a meaningful context. Files and hierarchical file systems have little relation with this, being more useful for the computer itself rather than for the users. To help users manage their files several approaches allow the visualization of the users' document collections in meaningful ways. Moving away from the file system hierarchies those approaches strive to convey an overall view of the users' documents.

Some approaches, such as, Treemaps [Johnson 91],

Cone Trees [Robertson 91], Collapsible Cylindrical Trees [Dachselt 01] and Fisheye Views [Furnas 86] try to cope with the simultaneous visualization of a large number of elements. They use colors to classify information, 3D spaces to highlight relevant information, and allow the navigation through hierarchical parameters or distribution of information in 3D spaces. Other systems allow us to visualize and browse through information in more specific domains, such as the WebBook [Card 96], the DeepDocument [Masoodian 04] and the Perspective Wall [Mackinlay 91], which explore concepts like document transparency and the book metaphor, to show the information in a specific context. Approaches developed by Wise [Wise 95], for visualization of large text documents, and by Rennison [Rennison 94], for visualizing large quantities of news stories, use the term galaxy in the context of information visualization. However, they did not explore the outer space metaphor as we are doing. These authors use the term galaxy to only express a large set of "points" arranged spatially, ignoring completely the way galaxies are composed and organized.

The simple visualization of files, in whatever way, is not enough to help users retrieving them. To actually retrieve a file, a file browser or "explorer" must be used. Unfortunately, most such solutions, again, fallback to the file system's conventions, making it difficult to use other information when retrieving documents.

Our solution tries to make use of the advantages of both

kinds of approach. It allows the visualization of the users' entire document collection, so that they might get an immediate feeling of its overall composition (numbers and types of documents, etc.). Documents are grouped by meaningful properties, such as their type or date, rather than a position in an arbitrary hierarchy. Outer Space is used as a metaphor for this representation. Different astronomical entities are used to represent documents and document collections. Rather than just allowing the visualization, users can navigate in this "document universe" searching for a particular document. As this is an interactive system, its interface takes a preponderant role. As such, a user-centered interface design approach was taken, ensuring the validity of the solution. On average, no document is more than four mouse clicks away.

In the following section we will describe in more detail how the Outer Space metaphor was used. Then, the low-fidelity prototypes that led us to our approach will be discussed, as will their heuristic evaluation. Next, we will present the system's architecture and a short description of the preliminary prototype. Finally, we will conclude with a mention of relevant future work.

2. Overview of Our Proposal

In our approach we use the metaphor of Outer Space. It is a domain the average user has become familiar with, due to exposure in games, literature and popular media. This familiarity will allow users to interact in a way such that the visualization itself will not interfere with the main tasks: finding documents. We focus on personal documents, with which the user has previously interacted.

The four most commonly known astronomical entities were used: galaxies, solar systems, planets, and satellites. The larger its real-world counterpart, the more documents it will contain. For instance, a galaxy might contain all song files, each solar system within can contain all songs from a single band, each planet a particular album, and its satellites are songs in that album.

At the beginning, the user is presented with the entire "universe": all galaxies (see Figure 1). It will then be possible to continuously zoom into the galaxies to reach the other entities. At each detail level, the documents are grouped according to some criterion. This can be chosen by the user using a menu. The different criteria reflect how users might remember their documents. For instance, each galaxy can contain all documents created in a year, or all documents of a certain type (music, images, etc.).

While users zoom in into the representation, document groups are shown with increasing detail. By being able to choose at each moment the criterion they feel more relevant, it is easy for users to conduct the browsing towards the desired documents. At no time are the users forced to choose specific criteria, being free to explore their document collections as they deem more appropriate for each document they seek. As the detail level increases, different criteria might become available. For instance, if at some point (as in the example above) the user is visualizing only

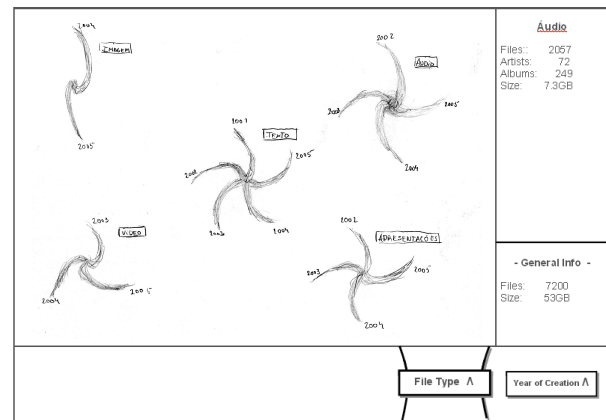


Figure 1. First low-fidelity prototype.

music files, it makes sense to group them by artist or album.

The data that underlies the documents' classifications according to the different criteria is automatically gathered beforehand, into a database. This is accomplished with the help of a sub-system that analyzes the users' hard drives' and collects not only information about the documents' names and whereabouts, but all meta-data it can glean from them. This includes actual meta-data in the documents (ID3 tags of mp3 files, for instance), but also keywords for text documents, generated with the help of the tfidf algorithm, etc. The goal is to gather a rich set of data about the documents with no need for user intervention, as few users would be willing to exhaustively classify all their documents in such a fashion.

The interface itself is generated based on that data. The different file types, dates, album names, etc. available to the user while browsing (and, even, the different criteria, such as file name, size, file type and so on) reflect those present in the database at the moment. The interface is, thus, flexible, adapting to each users' needs. It does not force them to deal with criteria and values they might find irrelevant, nor does it hinder them by preventing the use of relevant data.

At lower levels, when visualizing entities that might represent documents, we can use their properties to give them distinct visual appearances. Furthermore, we took care to ensure that those appearances and the location within the universe remain constant across sessions. This capitalizes on users' visual and spatial memories helping them to retrieve the same document a second time or related documents. Moreover, we take advantage of the several degrees of freedom to visually convey hints of the document's contents. For instance, if documents are planets, their size can represent the file size, the duration of a song or the number of pages of a text document, while their closeness to the star can symbolize the track order.

The actual visualization is generated from each particular document collection. For instance, if a galaxy contains documents from a particular year, each arm might repre-

sent a month or trimester. The actual granularity will be automatically decided by the system based on the number of documents to represent. If a galaxy representing audio files is organized by artist, and there are too many artists to represent each one as a galaxy arm then, the galaxy will appear as a sorted cloud of solar systems (an elliptic galaxy), rather than a spiral.

3. Low Fidelity Prototypes

The next step in our user-centered approach was to create a low-fidelity prototype using our Outer Space metaphor. Low fidelity prototypes are prototypes created in a short period of time (one to two hours), without writing a single line of code, using just paper, pencil and other low cost materials. Figure 1 presents an image of our first low-fidelity prototype, where we can see files organized by type in galaxies.

To validate our metaphor and our low-fidelity prototype, we performed a set of heuristic evaluations. We asked 4 usability experts (people who knew Nielsen’s usability heuristics [Nielsen 93] and were proficient in their use) to evaluate our prototype. We started by presenting the prototype to evaluators, explaining the concept and goal of the application and the tasks they can perform on it. After that, one person took notes of the evaluators’ comments while other person was playing the role of computer, simulating the behavior of the application. At the end of the evaluation, evaluators gave us a report summarizing the usability problems found. For each usability problem, they mentioned the heuristic(s) violated, the severity of the problem and a possible solution.

From this heuristic evaluation we concluded that the majority of the problems, if not all, were related to the menu at the bottom. All evaluators considered that the navigation was very difficult: i) They were not able to associate the selected option with the content of the main window; ii) Users could not identify the result of selecting each option of the menu; iii) It was not clear how to go back one step.

Taking these usability problems into account, we designed a second version of the low-fidelity prototype, where the main improvements were on the bottom menu and its navigation. Figure 2 shows the second low-fidelity prototype, where we can see the new menu. Before starting coding the first functional prototype, we performed another heuristic evaluation to the second version of the low-fidelity prototype. This time we asked four experts to evaluate our prototype. Two of them were evaluating the prototype for the first time, while the other two already evaluated the first version.

This second evaluation revealed that our prototype had a better usability than the first one. Evaluators identified less problems and they were less severe. The navigation was easier and all evaluators were able to complete the set of tasks. However, there still were some problems related to the correspondence between a menu option and the desired result. Also about the consequences of selecting a specific

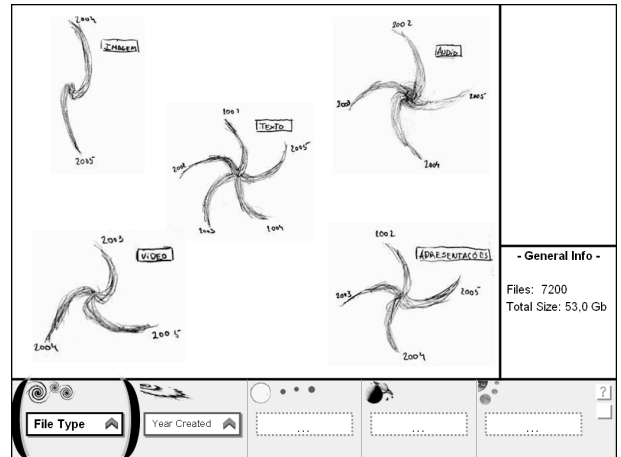


Figure 2. Second low-fidelity prototype.

option and finally, what are the areas in the main window that they could click on.

4. Architecture and Functional Prototype

Before starting developing the functional prototype, we defined the system architecture, depicted in Figure 3.

Our architecture is composed of three layers: Persistence Layer, Logical Layer and the Presentation Layer. The Persistence layer contains the database and the XML file with the configuration about the different attributes of files that can be used for the navigation and visualization of files. The Logical layer is responsible for creating and filling all the data structures according to the navigation steps performed by the user. Each time the user selects a type of visualization (by file type or by date) all data structures are rebuilt. In the Logical layer we have also all the elements from the Outer Space metaphor (galaxies, solar systems, etc.) filled with information. This way, when the system

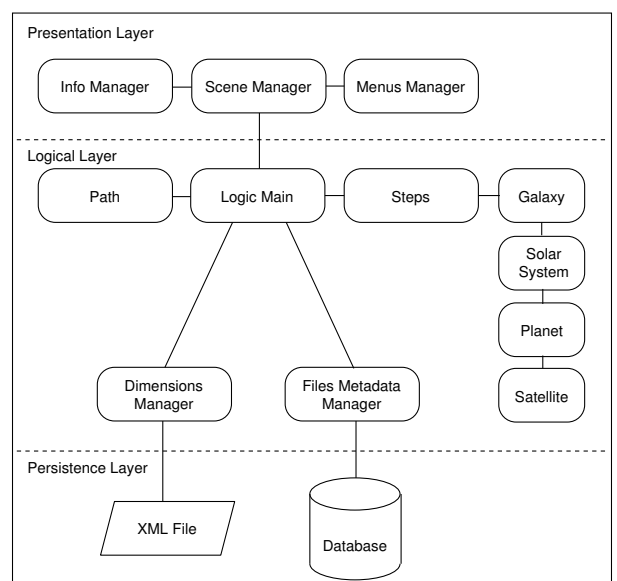


Figure 3. System Architecture.

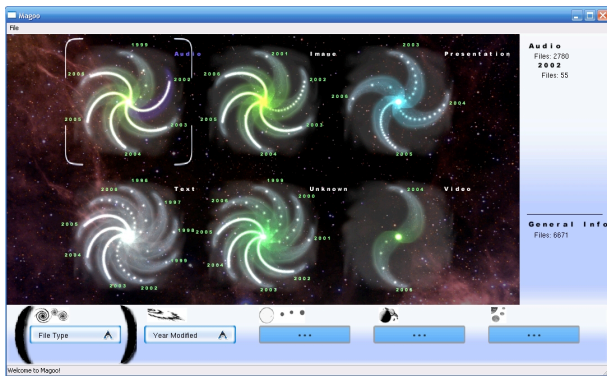


Figure 4. Functional prototype.

needs to represent one of the entities visually, only needs to access the information stored in the Logical layer. Finally, the Presentation layer is responsible for managing the navigation of the bottom menu, the scene graph and visually represent the different Outer Space entities.

We developed a preliminary functional prototype using this architecture, which at the time of this writing has very few functionalities (see Figure 4). In the near future we plan to conclude this functional prototype, submit it to another round of heuristic evaluation and improve the prototype by correcting identified problems.

5. Conclusions and Future Work

We have presented a new approach for document visualization and browsing based on the Outer Space metaphor. This solution has the advantage of allowing users to find a specific file in no more than four clicks. Moreover, users do not need to care about the directory where files were stored.

Currently, we have a very preliminary functional prototype that we are improving in terms of functionality and usability. To achieve this stage, we first created two low-fidelity prototypes, which were submitted to heuristic evaluations to identify usability problems.

After concluding the fully functional prototype, we plan to perform experimental evaluations with users, to measure its usability by comparing it to other “ordinary” file managers. In particular, we will measure the easiness that a particular file can be found, focusing on the number of mouse clicks and the time spent doing the task.

References

- [Card 96] Stuart K. Card, George G. Robertson, and William York. The webbook and the web forager: an information workspace for the world-wide web. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 111–ff., 1996.
- [Dachselt 01] Raimund Dachselt and Jürgen Ebert. Collapsible cylindrical trees: A fast hierarchical navigation technique. In *Proc. of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 79, 2001.
- [Furnas 86] George W. Furnas. Generalized fisheye views. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 16–23, 1986.
- [Johnson 91] Brian Johnson and Ben Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proc. of the 2nd conference on Visualization '91*, pages 284–291, 1991.
- [Mackinlay 91] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: detail and context smoothly integrated. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 173–176, 1991.
- [Masoodian 04] Masood Masoodian, Sam McKoy, Bill Rogers, and David Ware. Deepdocument: use of a multi-layered display to provide context awareness in text editing. In *Proc. of the working conference on Advanced visual interfaces*, pages 235–239, 2004.
- [Nielsen 93] Jakob Nielsen. *Usability Engineering*. Academic Press, 1993.
- [Rennison 94] Earl Rennison. Galaxy of news: an approach to visualizing and understanding expansive news landscapes. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 3–12, 1994.
- [Robertson 91] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 189–194, 1991.
- [Wise 95] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings of the 1995 IEEE Symposium on Information Visualization*, page 51, 1995.

IVY Trace Visualiser

Nuno Miguel Eira de Sousa José Creissac Campos
Departamento de Informática/CCTC
Universidade do Minho, Braga, Portugal
nunolesi@sapo.pt/josé.campos@di.uminho.pt

Sumário

No contexto do projecto IVY, tem vindo a ser desenvolvida uma ferramenta de modelação e análise de sistemas interactivos, tendo em vista a detecção de potenciais problemas de usabilidade no início do desenvolvimento de um qualquer sistema interactivo. Quando uma dada propriedade em análise não se verifica, a ferramenta procura indicar um contra-exemplo: um comportamento do modelo que demonstre a falsidade da propriedade em questão.

Estes contra-exemplos, no entanto, podem atingir tamanhos consideráveis, dependendo da complexidade do modelo, o que dificulta a sua análise. De forma a facilitar essa análise, a arquitectura da ferramenta IVY prevê um componente de suporte à análise. Este componente visa, através de representações visuais e de mecanismos de análise, facilitar a compreensão dos contra exemplos, de forma a tornar mais claro qual o problema que está a ser apontado e possíveis soluções para o mesmo.

Este artigo apresenta o componente de análise da ferramenta IVY. São apresentadas a arquitectura do componente, as representações implementadas e os mecanismos de análise disponibilizados.

Palavras-chave

Avaliação de sistemas interactivos, ferramentas de análise, *model checking*, representações visuais.

1. INTRODUÇÃO

No desenvolvimento de sistemas interactivos cruzam-se as áreas da Interação Humano-Computador (IHC) e da Engenharia de Software. Estudos mostram que o sucesso de tais sistemas depende, em grande medida, da sua usabilidade. A norma ISO DIS 9241-11 identifica os factores relevantes para a usabilidade de um sistema ao defini-la como a eficácia, eficiência e satisfação com que utilizadores determinados atingem objectivos determinados em ambientes específicos. Eficácia tem a haver com a possibilidade (ou não) de os utilizadores poderem atingir os seus objectivos utilizando o sistema num dado contexto. Eficiência tem a ver com o maior ou menor esforço que os utilizadores terão de despender para atingir esses objectivos. Satisfação é uma medida subjectiva do grau de agradabilidade na utilização do sistema.

O *Software Engineering Book of Knowledge* [SWEBOK01] considera a concepção de interfaces com o utilizador como uma área relacionada (mas distinta) da Engenharia do Software, não mencionando a área da IHC. Torna-se portanto necessário recorrer a peritos na área da IHC para ter em consideração factores de usabilidade no desenvolvimento de sistemas interactivos. Na prática, as áreas da Engenharia de Software e IHC têm vivido relativamente afastadas. Torna-se assim necessário procurar estabelecer pontes de contacto entre as duas áreas.

No contexto do projecto IVY (A model based development environment – POSC/EIA/56646/2004) procura-se desenvolver técnicas e ferramentas que facilitem a incorporação de preocupações de usabilidade no desenvolvi-

no desenvolvimento de software, promovendo a comunicação entre as duas comunidades. A abordagem seguida baseia-se na utilização de modelos para capturar aspectos relevantes dos sistemas interactivos e pretende possibilitar aos engenheiros de software uma maior autonomia na consideração de questões de usabilidade relacionadas com o comportamento do sistema, bem como identificar os pontos em que é necessário recorrer ao auxílio de peritos em IHC. O principal objectivo do projecto é o desenvolvimento de uma ferramenta de modelação e análise, tendo em vista a detecção de potenciais problemas de usabilidade no início do desenvolvimento de um qualquer sistema interactivo.

Os modelos são obtidos quer através de um processo de modelação baseado num editor, quer utilizando engenharia reversa no código da interface com o utilizador. A ferramenta visa permitir a verificação automática de modelos de sistemas interactivos, recorrendo ao *model-checker* SMV [McMillan93] para a análise do comportamento dos modelos.

Tal como será referido na secção 2, uma etapa importante do processo é a análise dos contra-exemplos produzidos pela ferramenta quando o teste de uma dada propriedade falha. Neste artigo apresenta-se a arquitectura e modo de funcionamento do componente responsável pelo suporte a este processo de análise. Na secção 2 apresentam-se alguns detalhes sobre a abordagem suportada pelo IVY, fornecendo deste modo contexto à discussão que se seguirá. Na secção 3 apresenta-se a arquitectura do componente de visualização. Na secção 4 apresentam-se

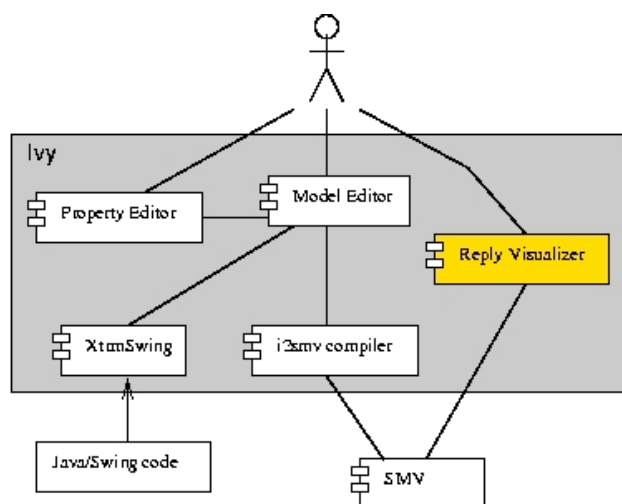


Figura 1. Arquitetura da ferramenta IVY

as diferentes representações visuais que têm vindo a ser exploradas. Na secção 5 apresentam-se os mecanismos de filtragem e marcação que permitem a exploração dos traços de comportamento. Na secção 6 apresenta-se um exemplo de aplicação da ferramenta. Finalmente, na secção 7 faz-se um resumo do trabalho realizado e lançam-se alguns apontadores para trabalho futuro.

2. IVY

A arquitetura da ferramenta é apresentada na figura 1. Tal como já descrito, o utilizador do IVY cria modelos quer através de um editor dedicado (model editor), quer através de um componente de engenharia reversa (XtrmSwing), quer ainda através de uma combinação das duas abordagens.

Os modelos são escritos recorrendo à linguagem MAL Interactors [Campos01, Campos04]. Um *interactor* [Duke93] pode ser visto como um objecto capaz de representar (parte d) o seu estado num qualquer meio de apresentação. Assim, um *interactor* possui um conjunto de atributos tipados que definem o seu estado e um conjunto de acções que definem as mensagens que o interactor pode receber e/ou enviar. No caso dos MAL interactores, o comportamento do *interactor* em reacção à mensagens recebidas é definido utilizando axiomas escritos na lógica MAL (*Modal-Action Logic*) [Ryan91]. Estes axiomas definem a semântica das acções em termos do seu efeito no estado do *interactor*.

Para a discussão que se segue, basta perceber que um modelo é construído compondo *interactors* de forma hierárquica. Deste modo, pode sempre ser representado por uma máquina de estados em que os estados são definidos pelos valores dos atributos e as transições são etiquetadas pelas acções que provocam as alterações nos atributos. A fase de verificação consiste em testar propriedades do comportamento dessa máquina de estados.

As propriedades para verificação são escritas em CTL (*Computational Tree Logic*) [Clarke86]. Para facilitar o processo de verificação, a ferramenta IVY disponibilizará um editor de propriedades (property editor), tendo em vista

```

Trace Description: CTL Counterexample
Trace Type: Counterexample
-> State: 1.1 <-
  others.action = nil
  others.visible = 0
  others.newinfo = 0
  others.mapped = 0
  mail.action = nil
  mail.visible = 0
  mail.newinfo = 0
  mail.mapped = 0
  action = nil
-> State: 1.2 <-
  others.action = map
  others.visible = 1
  others.mapped = 1
  mail.action = update
  mail.newinfo = 1
  
```

Figura 2. Traço de comportamento

permitir a definição de propriedades a verificar se ser necessário conhecer a linguagem lógica subjacente.

Para mais detalhes sobre a aplicação da lógica CTL no presente contexto ver [Campos01]. Para a discussão que se segue o importante é saber que, quando uma dada propriedade não se verifica, o SMV procura fornecer um traço de comportamento que demonstre a falsidade da propriedade em questão (ver figura 2 para um exemplo). Esse traço consiste numa sequência de estados da máquina que violam a propriedade.

Para possibilitar a sua verificação, os modelos MAL interactores são compilados para a linguagem do SMV (i2smv compiler). Uma vez que a expressividade da linguagem do SMV é limitada, quando comparada com a linguagem dos MAL interactores, o processo de compilação para SMV introduz uma série de variáveis auxiliares. Merece particular relevo a introdução de um atributo 'action', utilizado para modelar as acções, uma vez que o SMV não contempla esse conceito.

Para a discussão que se seguirá, um outro aspecto a merecer relevo tem a ver com os modelos de execução. Ao nível dos MAL interactores as acções dos diferentes interactores podem acontecer de forma assíncrona. Assim, um interactor pode executar uma acção enquanto os restantes se mantêm inactivos. Ao nível do SMV, no entanto, as transições de estado ocorrem de forma síncrona. Para modelar transições de estado assíncronas, torna-se necessário introduzir uma acção especial *nil* que ao nível dos MAL interactores (nível lógico) corresponde a nada acontecer, mas ao nível do SMV (nível físico) representa uma transição de estado (para um estado com os mesmos valores dos atributos). Deste modo, o módulo SMV correspondente a um interactor pode sofrer uma transição de estado associada a uma dada acção, enquanto os res-

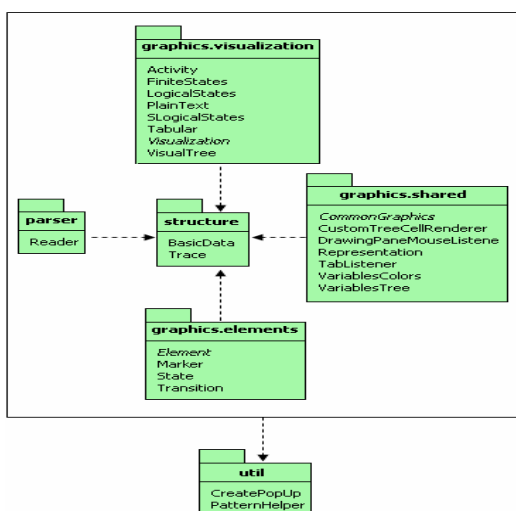


Figura 3. Arquitectura do visualizador

os restantes executam a transição associada a *nil* (ou seja, mantém o estado).

Os traços produzidos pelo processo de verificação fazem, tal como seria de esperar, referência às variáveis e estados existentes ao nível do código SMV. Assim, torna-se necessário reverter este processo para que as entidades referidas passem a ser as existentes ao nível do modelo original. Um exemplo típico será a eliminação do atributo ‘action’, substituindo-o pela indicação da acção nas transições entre estados.

Estes traços, no entanto, podem atingir tamanhos na ordem das dezenas de estados, dependendo da complexidade do modelo. De forma a facilitar a análise dos traços, a arquitectura da ferramenta IVY prevê um componente de análise de traços (reply visualizer – ver figura 1). Este componente de visualização visa, através de uma representação visual e de mecanismos de análise dos traços, facilitar a sua compreensão de forma a tornar mais claro qual o problema que está a ser apontado e possíveis soluções para o mesmo.

3. O VISUALIZADOR

Tal como já referido, o componente de visualização é responsável pela apresentação dos resultados do processo de verificação, tendo três responsabilidades principais:

- pré-processar os traços produzidos pelo SMV de forma a torná-los consistentes com a notação utilizada na escrita do modelo;
- fornecer representações visuais dos traços de forma a facilitar a sua compreensão;
- fornecer um mecanismo de análise dos traços.

A arquitectura do visualizador consiste em três componentes principais: Parser, Structure e Graphics [Sousa06] e um secundário: Util (ver figura 3).

O componente Parser é responsável por fazer o *parsing* do ficheiro do traço.

O componente Structure é responsável por guardar a informação, resultante do processo de *parsing* do ficheiro do traço, numa estrutura de dados.

O componente Graphics é responsável por visualizar a informação obtida pelos outros componentes, usando para isso diferentes representações visuais (por exemplo: diagrama de actividades, diagrama de estados finitos, etc.). Este é constituído por três sub-componentes:

- Visualization: responsável pelas representações visuais dos traços;
- Shared: responsável por fornecer funcionalidades comuns a todas as representações visuais;
- Elements: responsável pela implementação dos objectos gráficos que são usados pelas representações visuais, por exemplo: Transition (que pode ser desenhado como uma seta).

O visualizador (ver figura 4) permite ter disponíveis diversas vistas sobre o mesmo traço. Cada vista dá origem a uma *tab* e para cada *tab* pode ser seleccionada a representação visual que o utilizador desejar através de uma *combobox* com as opções possíveis.

O componente Util fornece serviços que são usados pelos componentes principais do visualizador.

Na parte superior da janela do visualizador é apresentada a fórmula em análise na notação CTL. Como o traço foi gerado, a fórmula é falsa. O objectivo mais imediato que um utilizador IVY querará concretizar, com a ajuda da ferramenta, será o de descobrir o estado (ou estados) que violam a fórmula. Depois disso, um outro objectivo será compreender as razões dessa falha, analisando os atributos dos *interactors* e as transições entre estados. Esta análise é efectuada recorrendo às funcionalidades de filtros e marcações que serão descritas mais à frente neste artigo.

4. REPRESENTAÇÕES VISUAIS

Actualmente o visualizador possui implementadas as seguintes seis representações visuais:

- *Trace* – a representação textual original produzida pelo SMV;
- *Tree* – representação em árvore dos estados do traço;
- *Tabular* – representação tabular semelhante à existente no SMV da Cadence Labs;
- *Fisical States* – representação gráfica dos estados do traço;
- *Logical States* – representação semelhante à anterior em que os estados do traço são pré-processados para eliminar estados artificiais introduzidos pelo processo de compilação;
- *Activity Diagram* – representação centrada nas acções recorrendo a diagramas de Actividade do UML 2.0 [OMG05] (para uma introdução ao UML ver [Fowler04]).

Cada uma delas será agora descrita em mais detalhe.

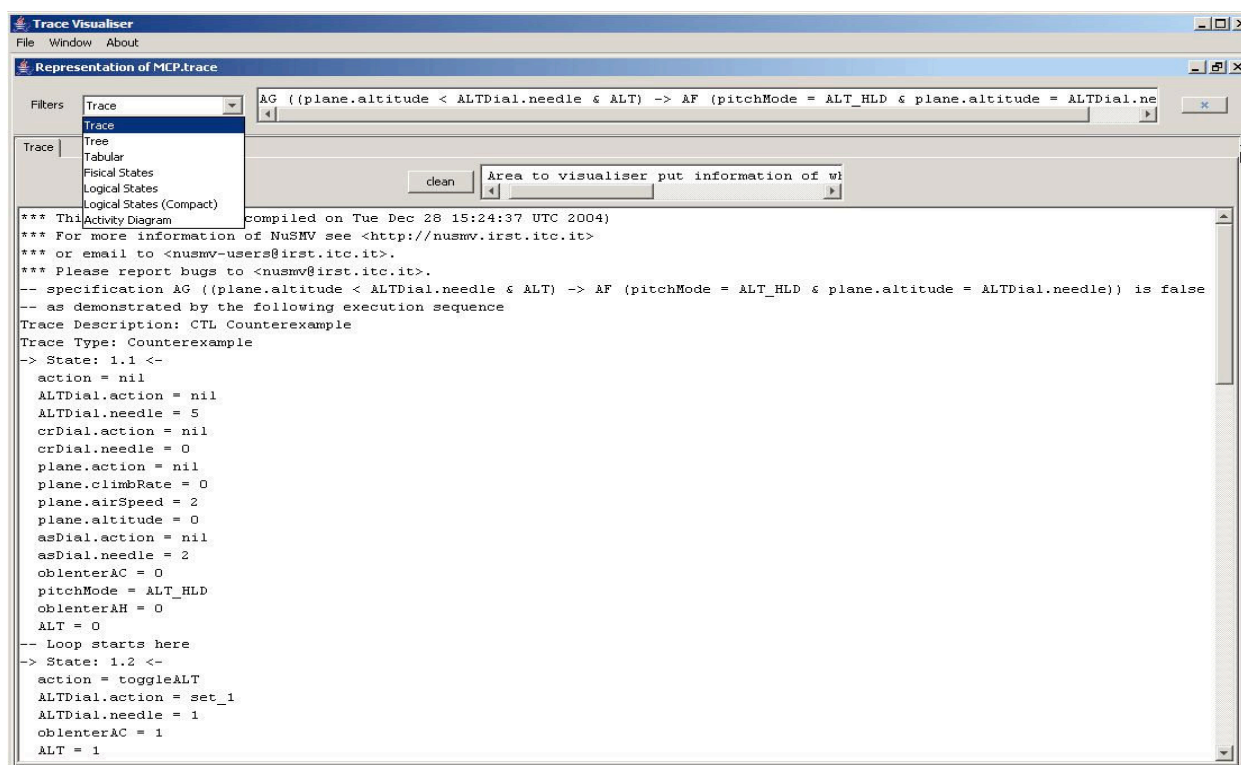


Figura 4. Interface Gráfica do Visualizador mostrando o ficheiro de trace original

4.1 Trace

A representação *Trace* (ver figura 4) corresponde ao formato textual do traço que serve de *input* ao visualizador, tal como produzido pelo SMV.

Um traço é uma sequência de estados, cada um deles definido à custa dos atributos dos *interactors* do modelo. No traço só os atributos de *interactors* que mudaram são mostrados em cada estado. Os atributos são representados na seguinte notação: <interactor>.<atributo>. Quando um atributo surge sem o prefixo do *interactor* isso significa que pertence ao *interactor main* (o *interactor* raiz do modelo).

Tal como referido anteriormente, o atributo ‘action’ modela as acções e corresponde à acção que levou o *interactor* para o estado corrente.

Antes de um estado pode existir uma indicação “Loop starts here” que indica a existência de um ciclo: chegando-se ao estado final do traço, este continua no estado em que a indicação surge.

Esta representação apresenta o traço sem qualquer tipo de pré-processamento. Assim, a sua interpretação requer alguns conhecimentos relativos ao processo de compilação. Adicionalmente, é uma representação que tende a ser extensa e difícil de analisar.

4.2 Tree

Tree é uma representação visual em forma de árvore e uma versão mais estruturada da representação anterior.

Nesta representação (ver figura 5), cada um dos nodos de topo da árvore (a seguir ao nodo root) corresponde a um estado do traço. Dentro de cada estado, os atributos e respectivos valores são agrupados, por *interactor*, nos

nodos inferiores da árvore. Assim, o *interactor main*, implícito na representação anterior, surge aqui de uma forma explícita.

O facto de a informação dos estados ser representada numa árvore permite, comparativamente com a representação original, acrescentar novas funcionalidades, dado que o componente *JTree* (a implementação da árvore) possui uma série de características base úteis. Essas características são as seguintes:

- a possibilidade de representar relações de inclusão, agrupando folhas em nodos (neste caso as folhas são os atributos e respectivos valores e os nodos são *interactors/estados*);
- a possibilidade de expandir e colapsar nodos (neste caso, representando estados do traço ou *interactors*), permitindo ajustar em cada momento a informação que é apresentada;
- a possibilidade de fazer o *rendering* (cor do texto, cor de fundo, etc) de um nodo da árvore de forma diferenciada (neste caso os atributos com texto de cor vermelha e as acções com texto de cor azul).

A utilidade desta representação reside na possibilidade de apresentar a informação de forma agrupada e no facto de permitir seleccionar apenas a informação que se deseja, por exemplo ver só determinados estados (colapsando os outros restantes) ou ver só alguns *interactors* dentro de um estado.

Esta representação continua a sofrer de alguns dos problemas na representação original, pelo que poderá não

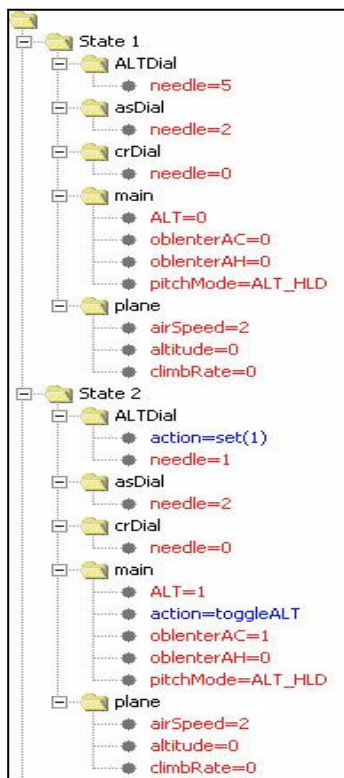


Figura 5. Representação visual Tree

ser das mais úteis na fase de análise. As principais limitações advêm do facto de ainda ser uma representação extensa e de as acções não serem representadas de um forma muito diferenciada relativamente aos atributos dos *interactors*. Por exemplo: uma acção representada por uma seta fornece uma melhor intuição de transição. No entanto, quando comparada com a representação original, a representação Tree possibilita uma visão mais estruturada do traço, pelo que poderá servir de complemento às representações visuais que serão apresentadas de seguida.

4.3 Tabular

Nesta representação (ver figura 6) a informação é apresentada numa tabela ao estilo das representações fornecidas pelo SMV da Cadence Labs ou por [Loer04]. Os cabeçalhos da coluna mostram o número dos estados. O início de um ciclo é mostrado usando um asterisco.

Uma célula com fundo a branco mostra que o valor do atributo nesse estado mudou quando comparado com o valor do atributo no estado anterior. Quando o valor se mantém de um estado para outro o fundo da célula é mostrado a cinzento. Esta ideia, adoptada de [Loer04], é muito relevante pois permite saber quando é que os *interactors* mantêm o seu estado e também quando é que (ao nível lógico) estes sofrem uma transição do mesmo.

A representação tabular é uma das de mais fácil compreensão, provavelmente por ser uma das mais conhecidas e usuais. No entanto não deixa de ter um grande inconveniente: dado o elevado número de estados que os traços poderão conter, não é muito prático e eficaz estar a analisar a informação neles contida por meio de tabelas,

	1	2*	3	4
ALTDial.action		set(1)	set(2)	
ALTDial.needle	5	1	2	2
asDial.action				
asDial.needle	2	2	2	2
crDial.action				
crDial.needle	0	0	0	0
main.ALT	0	1	1	0
main.action		toggleALT		enterAC
main.oblenterAC	0	1	1	0
main.oblenterAH	0	0	0	0
main.pitchMode	ALT_HLD	ALT_HLD	ALT_HLD	ALT_CAP
plane.action				fly
plane.airSpeed	2	2	2	5
plane.altitude	0	0	0	1
plane.climbRate	0	0	0	1

Figura 6. Representação visual Tabular

tabelas, pois rapidamente esta se torna difícil. A utilização de cor para marcar as células com alteração de informação relativamente ao estado anterior, bem como a possibilidade de utilização de filtros e marcações (ver secção 5) permitem uma expressividade adicional que ajuda a obviar este problema.

A representação tem a seu favor a compactação de informação que proporciona, permitindo uma visão global imediata de toda a informação se o número de estados não for demasiado elevado. Assim, pode ser útil na análise de traços com poucos estados, ou como complemento de outras representações no caso de traços com um número elevado de estados. Tem ainda a vantagem de ser uma representação familiar a utilizadores do *model checker*.

4.4 Físical States

Esta representação inspira-se nos diagramas de transição de estado. Para cada *interactor* (ver figura 7) existe uma coluna mostrando os estados que o interactor atravessa ao longo do traço. O estado global, com todas as variáveis dos *interactors*, é também representado (coluna GLOBAL) para servir como índice. O índice é usado para ver em cada momento o estado global formado pelo conjunto dos estados de cada *interactor*, bem como para ver a ordem dos estados. Neste estado global também pode existir uma seta de cor verde indicando o início de um ciclo e outra seta indicando o seu fim no último estado, caso um ciclo exista no traço.

No caso das colunas relativas aos *interactors*, os atributos e os seus valores são mostrados junto ao rectângulo que representa o estado. As acções são mostradas como setas entre dois estados consecutivos com uma etiqueta com a identificação da acção. Essa seta só é mostrada se existir uma acção para o *interactor* no estado respectivo. Quando não é mostrada isso significa que a transição se ficou a dever à acção *nil* (o estado desse *interactor* não mudou, mas um outro qualquer *interactor* sofreu uma transição).

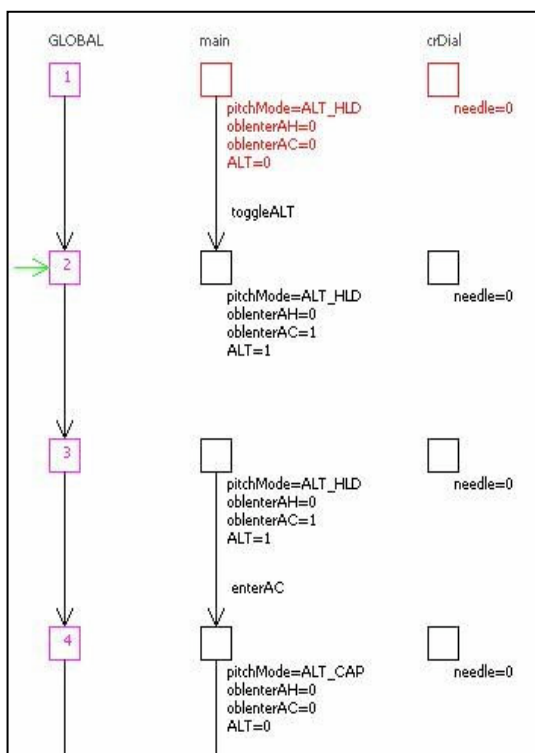


Figura 7. Representação visual Fisical States

No caso do estado global, a seta é sempre mostrada mas sem a etiqueta da acção. Os atributos não são mostrados directamente no diagrama, mas apenas quando se coloca o rato sobre um estado.

O utilizador pode decidir se pretende que a informação sobre os estados seja ou não apresentada directamente no diagrama, utilizando a funcionalidade de *popups*.

A funcionalidade dos *popups* é bastante útil porque permite seleccionar a informação que se pretende ver. Por exemplo, imaginando que se querem ver apenas as transições entre estados contidas no traço, só é necessário activar os *popups* e desta forma só são mostradas as acções. Se o utilizador quiser ver os atributos de um determinado estado basta colocar o rato dentro do rectângulo que o representa e a informação pretendida surgirá. O facto da informação não surgir toda de uma vez no ecrã (com *popups* activado) ajuda o utilizador a diminuir a quantidade da informação que é necessário analisar para descobrir o problema presente no traço.

A representação dos interactores em colunas permite separá-los uns dos outros e ver a sequência de estados de cada um de forma individual. No entanto isto levanta um problema em termos de informação redundante, dado que, em muitos casos, não existe verdadeira alteração do estado do interactor, mas sim a necessidade de transitar para outro estado por causa de uma transição ocorrida num outro interactor (ver explicação sobre estados físicos e estados lógicos na secção 2). Esta questão é eficazmente tratada noutra representação (Logical States) que irá ser descrita na próxima secção.

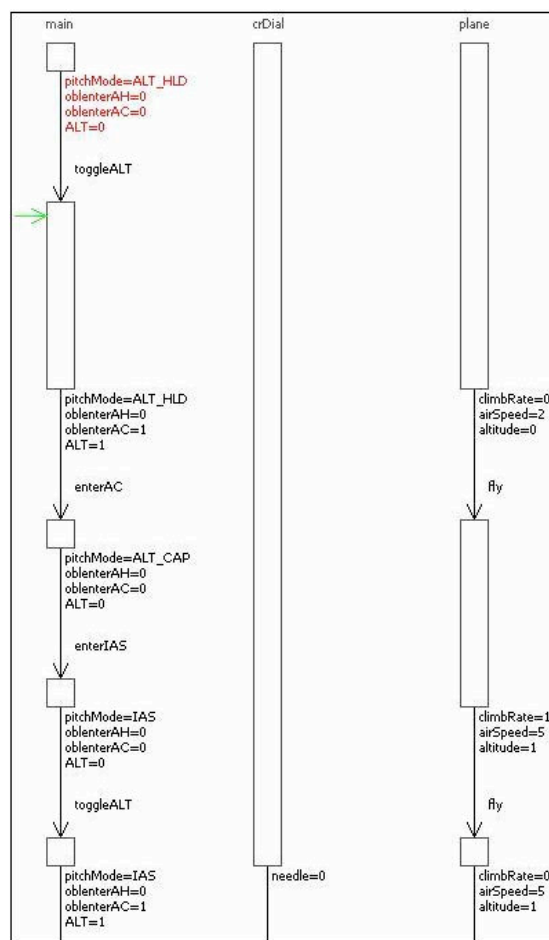


Figura 8. Representação visual Logical States

A representação Fisical States será potencialmente uma das mais úteis na análise devido à sua estruturação dos *interactors* por colunas, funcionalidades dos *popups* e representação das acções por setas. No entanto tem alguma redundância de informação que, tal como referido, será corrigida na representação visual Logical States.

4.5 Logical States

Esta representação (ver figura 8) é semelhante à anterior com a diferença que são representados os estados lógicos e não os estados físicos. Deste modo desaparecem sequências de estados sem transições entre eles. Em vez disso, é apresentado um único estado que “cobre” todos os estados físicos que acontecem ao nível do SMV.

Esta representação é útil para mostrar as mudanças de estado reais que acontecem no modelo, eliminando a redundância que acontecia na representação anterior. No entanto, ao fazer-se a análise do traço pode acontecer uma certa confusão quando se quer comparar *interactors* ao nível de um estado. O que acontece é que o estado de um dado *interactor* pode prolongar-se por vários estados de um outro interactor. Uma vez que os valores dos atributos de um estado só são mostrados no fim do mesmo, a ideia de cada linha do *diagrama* conter o mesmo estado para todos os *interactors* fica um pouco desvirtuada em termos gráficos.

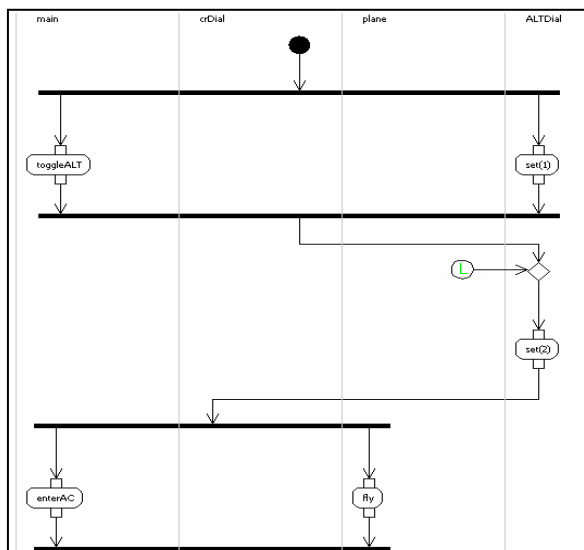


Figura 9. Representação Visual Activity Diagram

Apesar disso, graficamente esta representação tem um aspecto muito apelativo e pode fornecer inicialmente mais informação visual ao utilizador. Isto acontece devido à criação dos rectângulos grandes que eliminam a redundância da visualização do estado do *interactor*, quando na verdade este se mantém constante. Apenas se mostra o estado uma vez em vez de n .

Nesta representação o índice da representação anterior deixou de existir, preferindo-se mostrar o estado global como *label*.

4.6 Activity Diagram

Esta representação (ver figura 9) segue a notação da UML 2.0 para diagramas de actividade. Sendo o UML a norma para modelação orientada a objectos, é de esperar que os engenheiros de software conheçam e facilmente compreendam a notação. Adicionalmente, sendo os diagramas de actividade baseados em diagramas de fluxo de dados, é de esperar que mesmos peritos da área de IHC não tenham grande dificuldade em perceber a notação. As actividades são representadas por um rectângulo com os cantos arredondados. Os pequenos rectângulos associados às actividades representam os estados do *interactor* antes e depois das actividades ocorrerem. Os valores dos atributos podem ser consultados através de um *popup* colocando o rato sobre os rectângulos representativos dos estados.

Esta representação é bastante útil pois permite ver, de uma forma rápida, o resultado que a acção (actividade) produziu no estado de um *interactor*. Também é possível ver o estado global colocando o rato sobre a linha que liga duas actividades. A barra de sincronização de actividades permite ver as acções que são efectuadas em paralelo.

Como esta representação tem um claro foco nas acções, os atributos dos *interactors* são de certa forma escondidos, surgindo apenas como *popups*, caso essa opção

opção esteja activada. Uma análise de validade de uma fórmula, com maior ênfase concentração nas transições entre estados (acções), pode utilizar esta representação pois para esse efeito é a mais aconselhada.

Esta representação pode também ser a mais aconselhada quando existem interactores com muitos atributos porque não os mostra directamente, apenas como popups, o que proporciona uma maior compactação. Pelo contrário isto também pode ser uma desvantagem porque é necessário fazer a inspecção individual dos estados dos interactores que se deseja analisar, usando o rato, o que pode tornar-se desagradável quando existem muitos estados no traço. Para minorar este problema existem as funcionalidades dos filtros e marcações que serão descritas no próximo capítulo.

Uma aproximação à representação ideal pode ser conseguida com a utilização de todas as representações descritas nesta secção. Cada uma delas tem vantagens e desvantagens, por isso é necessário usar o que cada uma tem de melhor e conjugar tudo isso para se conseguirem atingir os objectivos desejados. Neste caso, eles são: descobrir o(s) estado(s) onde a fórmula do traço não se verifica e posteriormente compreender porque é que isso acontece.

5. FILTROS E MARCAÇÕES

5.1 Filtros

De forma a facilitar a análise dos traços produzidos pelo SMV, o visualizador permite que em cada *tab* se possa aplicar um filtro que corresponde a fazer uma selecção de estados com base num determinado critério referente aos atributos dos interactores. Sempre que se aplica um filtro desse tipo é adicionado mais um *tab* com os resultados desse filtro à barra de *tabs*. Depois é possível aplicar novamente outro filtro a esse *tab* resultante (*subfiltering*) e também escolher a representação visual que se desejar. O caso limite em que o filtro já não gera mais *tabs* acontece quando o resultado do filtro associado ao *tab* só contém um estado.

Os resultados dos filtros são apresentados colorindo os elementos gráficos relevantes tais como nodos das árvores, linhas e colunas de tabelas, ou setas; em função da representação visual específica.

Na representação Tree, por exemplo, quando são usados filtros que devolvem estados, os nodos dos estados presentes no resultado são expandidos e os outros colapsados de forma a destacar só os estados pretendidos. Quando são usados filtros que devolvem variáveis de *interactors*, as células da árvore que correspondem a essas variáveis são coloridas com a cor magenta.

Adicionalmente, algumas representações visuais possuem uma funcionalidade de animação simples, por exemplo, colorindo em sequência os estados na representação visual Fisical States.

Os filtros disponíveis no visualizador são os seguintes:

1. Retornar todas as acções de um estado;

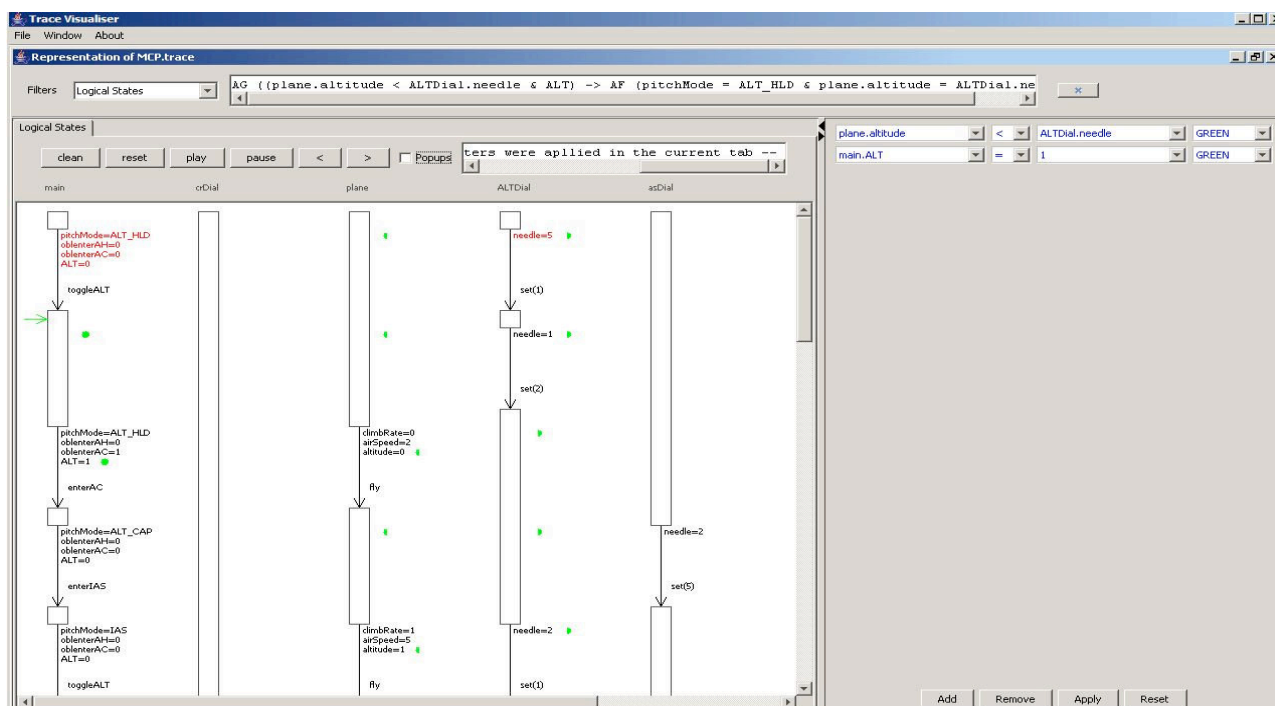


Figura 10. Análise do exemplo no diagrama de Físical States usando marcadores

- Retornar todos os estados onde pelo menos um atributo de uma selecção de atributos tem determinado valor;
- Retornar todos os estados onde todos os atributos de uma selecção têm determinados valores;
- Retornar todos os estados onde pelo menos um atributo de uma selecção de atributos mudou para um determinado valor;
- Retornar todos os estados onde todos os atributos de uma selecção mudaram para determinados valores;
- Retornar todos os valores dos atributos de interactors num estado;
- Retornar todos os valores para um atributo de um interactor.

A escolha dos parâmetros para os filtros é feita através de um *Wizard*. Num primeiro painel o utilizador escolhe as variáveis dos *interactors* que pretende analisar, num segundo painel escolhe os valores que deseja para estas.

5.2 Marcadores

Existe também a possibilidade de marcar estados em função de critérios definidos sobre os atributos dos estados. Neste caso os critérios são definidos num painel presente directamente na janela principal (ver figura 10).

Os critérios são definidos estabelecendo relações (=, >, <) entre pares de atributos ou entre atributos e valores. A cada critério é associada uma cor e todos os estados que verifiquem um dado critério são anotados com a cor relativa ao mesmo.

No caso de se efectuar uma comparação entre atributos, são desenhados dois semicírculos a cheio, com a cor

escolhida. Cada semicírculo é desenhado junto de cada um dos atributos, para desta forma os relacionar através da condição. No caso de comparações entre atributos e valores, são desenhados círculos a cheio com a cor escolhida. Se a opção dos *popups* estiver activada pode ser consultada a condição representada por cada marcador colocando o rato sobre o mesmo.

6. UM EXEMPLO

Para ilustrar a utilização da ferramenta iremos aplicá-la a um exemplo apresentado em [Campos01]. Trata-se da análise de um modelo do painel de controlo do piloto automático do avião MD-88.

Não sendo viável, nem relevante, apresentar aqui todo o exemplo, iremos restringir-nos à fase de análise. Após edição e compilação do modelo, uma das propriedades que se pretendeu verificar no modelo era a seguinte: sempre que o piloto automático está programado para atingir uma dada altitude, o avião acabará por voar a essa altitude em modo de manutenção da mesma. Em termos de usabilidade, esta propriedade pode ser relacionada com aspectos de previsibilidade do sistema, que influenciam o modo como os utilizadores conseguem antecipar a resposta do sistema e aprender a utilizá-lo. A propriedade pode ser expressa pela fórmula CTL:

$$AG(\text{plane.altitude} < \text{AltDial.needle} \ \& \ \text{ALT} \ \rightarrow \ \text{AF}(\text{plane.altitude} = \text{AltDial.needle} \ \& \ \text{pitchmode} = \text{ALT_HLD}))$$

(note-se que a ferramenta IVY irá disponibilizar um editor de propriedades que permitirá escreve-las sem recorrer directamente a CTL).

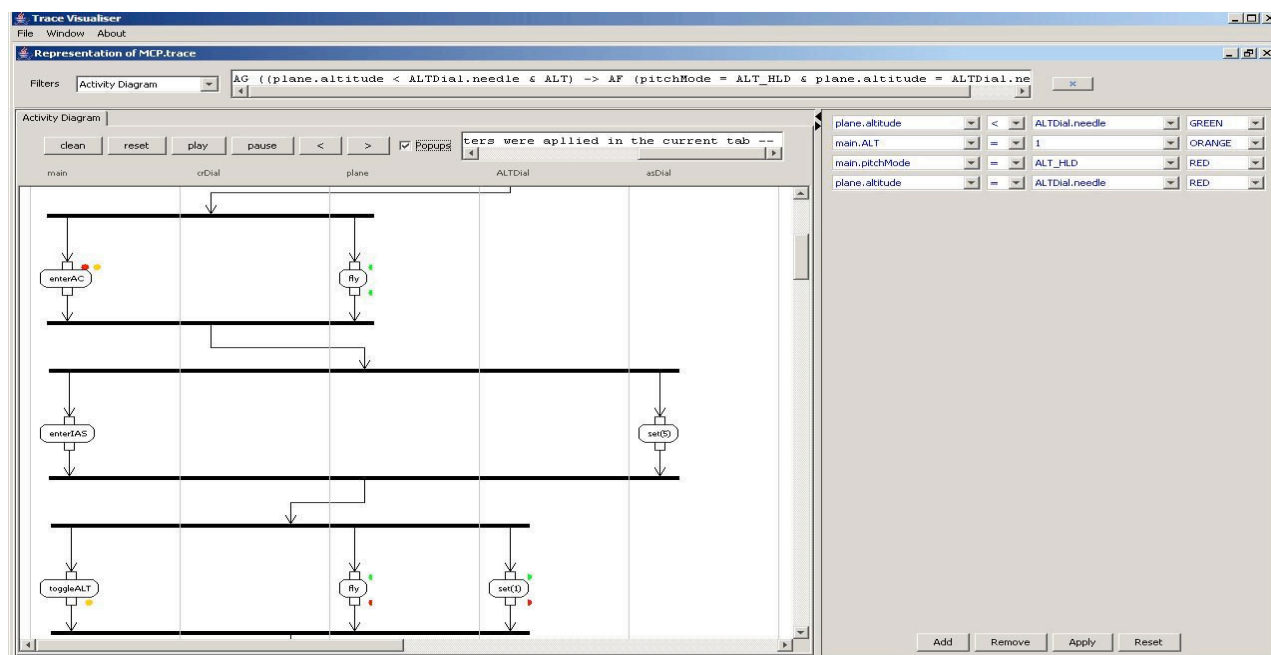


Figura 11. Análise do exemplo na representação Activity Diagrams usando marcadores

Quando se tenta a verificação da fórmula o SMV informa que ela é falsa e produz um contra exemplo (neste caso com 13 estados). Iremos agora, utilizando o visualiser, tentar perceber qual o problema que está a ser indicado pelo traço. Numa primeira fase procurou-se identificar quais os estados que verificavam o pressuposto da propriedade (avião programado para atingir uma dada altitude – $plane.altitude < AltDial.needle \ \& \ ALT$). Optamos por criar critérios de marcação para as sub-fórmulas desta conjunção e associar-lhes a cor verde.

O resultado da aplicação da marcação mostrou que a condição passa a existir logo no segundo estado. A melhor representação para constatar este facto revelou ser a baseada em diagramas de estado. A figura 10 mostra o diagrama resultante. No estado inicial do modelo (ver topo do diagrama) apenas aparecem os semicírculos resultantes do primeiro critério (que relaciona dois atributos). Após a acção `toggleALT`, a marcação passa a conter não só os semicírculos do primeiro critério, como o círculo relativo ao segundo.

De seguida procurou-se confirmar que de facto um estado com o avião estabilizado à altura pretendida não acontece no traço. Criaram-se duas novas marcações para as expressões

$$\begin{aligned} & plane.altitude = AltDial.needle \\ & \text{e} \\ & pitchmode = ALT_HLD \end{aligned}$$

tendo-se associado a cor vermelha a ambas. Tal como seria de esperar nenhum estado ficou anotado com duas marcas vermelhas (sinal que a conjunção das duas expressões não ocorre no traço).

Finalmente, para procurar perceber a razão para o sucedido, investigou-se o atributo ALT, que modela o facto de a captura de altitude do piloto automático estar armada. Para o efeito alterou-se a cor do critério que lhe estava

estava associado para laranja, de forma a distingui-lo do critério relativo à altitude.

Neste caso optou-se por um Diagrama de Actividade (ver figura 11). A análise das marcações resultantes chamou a atenção para o que se passa após o evento `enterAC` (evento que ocorre de modo automático e que é responsável pela activação de um modo intermédio – `ALT_CAP` – de aproximação final à altitude desejada). Com efeito, após esse evento, a captura de altitude (ALT) é desligada, apesar de o avião ainda estar em fase de aproximação à altitude programada (a marcação laranja desaparece, mas as marcações vermelhas não estão presentes). O que o traço mostra é que se nesse momento for alterada a velocidade vertical (evento `set(5)`) o piloto automático muda para modo de manutenção de velocidade vertical, perdendo-se o modo `ALT_CAP` e a captura de altitude. A partir desse momento é possível que o avião ultrapasse a altitude inicialmente programada no piloto automático (ver figura 11) pois o piloto automático já não está programado para parar na altitude indicada no mostrador.

7. CONCLUSÕES

Neste artigo foi apresentado componente da ferramenta IVY responsável por fornecer mecanismos de visualização e análise de traços de comportamento gerados pelo *model checker* SMV. Para além dos mecanismos de visualização, o componente fornece ainda mecanismos de análise or forma a auxiliar a identificação dos problemas encontrados pelo *model checker*. Neste momento todas as representações visuais descritas estão funcionais.

A utilização da ferramenta tem mostrado grande potencial na facilitação da análise dos contra exemplos fornecidos pelo SMV. A funcionalidade fornecida pelos marcadores, em particular, é muito promissora quando combinada

quer com diagramas de actividade quer com as representações baseadas em estados.

Tendo em vista avaliar a utilidade de cada tipo de representação de modo mais rigoroso, está planeado um estudo com utilizadores em que se fará uma avaliação comparativa de cada uma no suporte à análise dos traços de comportamento.

O trabalho na ferramenta irá ainda continuar explorando diversos aspectos com vista a melhorar a ferramenta:

- Explorar novas representações, por exemplo para permitir animações mais elaboradas (um aspecto que aqui ficou apenas mencionado).
- Melhorar os marcadores, para que permitam condições mais complexas (por exemplo, permitindo relacionar atributos em estados diferentes).
- Melhorar a representação tabular no seguinte aspecto: na coluna dos atributos dos interactores o conceito de interactor não está claramente presente; os atributos são apresentados todos ao mesmo nível, com a estruturação em interactores indicada da forma <Interactor>.<Atributo>. Seria útil poder agrupar explicitamente os atributos por *interactor*, indicando para cada um todos os seus atributos de forma estruturada na tabela.

8. AGRADECIMENTOS

Este trabalho é suportado pela FCT (Portugal) e pelo FEDER (União Europeia) no âmbito do contrato POSC/EIA/56646/2004.

Os autores agradecem a Alexander de Ridder e Francisco Martinez Posadas o trabalho realizado em versões iniciais da ferramenta [Ridder05]. Agradecem ainda a António R. Fernandes a sugestão de criação dos marcadores.

9. REFERÊNCIAS

- [Campos01] J. C. Campos, M. D. Harrison. Model Checking Interactor Specifications. *Automated Software Engineering*, 8(3-4): 275-310, Agosto, 2001.
- [Campos04] J. Creissac Campos, *Análise de usabilidade baseada em modelos*, Interacção 2004 – 1ª. Conferência Nacional em Interacção Pessoa-Máquina, 171-176, Grupo Português de Computação Gráfica, Julho 2004.
- [Clarke86] E. M. Clarke, E. A. Emerson, A. P. Sistla. Automatic Verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2): 244-263, 1986.
- [Duke93] D. J. Duke, M. D. Harrison. Abstract Interaction Objects. *Computer Graphics Forum*, 12(3), 25-36, 1993.
- [Fowler04] M. Fowler. *UML Distilled, third edition*. Object Technology Series, Addison-Wesley, 2004.
- [Loer04] K. Loer, M. Harrison. *A Framework and Supporting Tool for the Model-based Analysis for Dependable Interactive Systems in the Context of Industrial Design*. Technical report CS-TR-873, School of Computing, University of Newcastle upon Tyne, November, 2004.
- [McMillan93] K. L. McMillan. *Symbolic Model Checking: An Approach to the State Explosion Problem*, Kluwer Academic, 1993.
- [OMG05] Object Management Group, *Unified Modeling Language: Superstructure, v. 2.0*. OMG specification: formal/05-07-04, Agosto, 2005.
- [Ridder05] A. de Ridder, F. M. Posadas, J. C. Campos. *Technical guide for the Visualizer Component*. IVY technical report IVY-TR-5-01, Junho, 2005.
- [Ryan91] M. Ryan, J. Fiadeiro, T. Maibaum. Sharing actions and attributes in modal action logic. *Theoretical Aspects of Computer Science*, vol. 526 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 569-593, 1991.
- [Sousa06] Sousa, Nuno M.E., *IVY Trace Visualiser*, Relatório de Opção III, DI/UM, Fevereiro 2006.
- [SWEBOK01] *Guide to Software Engineering Book of Knowledge*, trial version 1.0 . IEEE, Maio 2001.

I-UsE: Integrated Usability Evaluation Environment

Luís Marcelino Vasco Amaral
DI-FCT-UNL
Quinta da Torre, Caparica
{LMarcelino,Vasco.Amaral}@di.fct.unl.pt

Abstract

Software usability evaluation (UE) is a typically neglected phase by software developers. Due to the high costs of properly done formal usability studies, companies easily cut investment, contributing deliberately to the development of biased software with usability problems. Given this panorama, the development of UE tools is becoming increasingly important in order to tackle both the costs associated with the collection of data and the analysis of usability experiments.

This paper presents a practical software tool to support formal studies of UE by combining video recording, live screen capture and analysis features together. This open tool is being developed using standard hardware and free software components, making it a cost effective solution for UE.

Keywords

Usability Evaluation, HCI, Multimedia, User testing, Usability tools.

1. INTRODUCTION

The importance of good software usability is widely recognized and is even subject to quality regulation. For example, the standard ISO 9241 is a document with seventeen parts that specifies requirements of “Ergonomics of Human System Interaction”. It is widely accepted that to achieve good usability, the software development process must include evaluation [Gediga02].

Despite the importance of software usability, it is often left to a secondary plan. Companies that assess their software’s usability tend to commit few resources to those studies. Frequently, there is one single person responsible for testing the software, from finding bugs to find usability problems. As hiring adequate experts is costly, these tests are typically performed by one of the programmers that developed the software, contributing to biased and wrongly validated software.

For usability evaluations (UE) involving users, the percentage of problems identified increases with user’s sample size [Faulkner03]. However, financial and temporal cost of the study also increases with the number of users involved in the study. This may be one of the reasons why companies commit so few resources to assess their software: the expense associated with more formal studies.

The expense in terms of financial and temporal resources of a usability study has two significant contributions: the cost of preparing and conducting the experiment and the cost of analysing the gathered data. The cost associated with the conduction of the experiment may be difficult to

minimise, as it depends on the type and number of users involved. However, there are different strategies that may reduce the cost associated with the analysis of a usability experiment. An example is the use of automated tools to analyse collected data.

Our solution combines both recording of data to be collected during the usability experiments and analysis features. The presented solution adopts semi-automated usability evaluation methods, some of which already identified in the literature.

The next section presents some of these usability evaluation methods. Section 3 describes some projects aimed at automating the usability assessment process. The requirements for a tool able to assist generic usability studies are identified in Section 4. Section 5 includes details of the proposed platform and section 6 presents some of the early conclusions and outlines future developments for the presented solution.

2. EVALUATION METHODS

Results of usability evaluations can vary widely when studying the same interface. Molich et al. found that the identification of usability problems is not consistent across usability teams [Molich04]. Furthermore, for studies conducted by the same team with samples of 5 users, Faulkner reported that the same experiment might report from 55% to nearly all usability problems [Faulkner03]. As suggested by Ivory and Hearst, the automation of some aspects of usability evaluation is a solution to achieve systematised results [Ivory01].

Automation on usability testing methods has been used predominantly in two ways: automated capture of use data and automated analysis of log files according to some metric. In a survey about the state of the art in automating usability evaluation, Ivory and Hearst present a taxonomy for usability evaluation models [Ivory01]. This taxonomy of UE classifies five method classes and within these classifies several method types. Two of the method classes are particularly relevant to this discussion: testing and enquiry.

The high number of automation approaches of log file analysis is a justification for Hilbert and Redmiles' survey about techniques to extract usability information from Interface Events. [Hilbert00]. They identify high level categories of techniques that emerged from their survey of different approaches:

- Synchronization and searching;
- Transformation;
- Analysis;
- Visualisation;
- Integrated Support.

3. EVALUATION TOOLS

This section reviews some of the tools available to support usability evaluations (UE), sometimes referenced as CAUSE-Tools (Computer-Aided Usability Engineering). The software tools presented may favour particular evaluation methods, some of which are not used in the proposed software solution.

3.1 WCAG Verification tools

There are several tools to test and validate Web sites according to the Web Content Accessibility Guidelines (WCAG) [Chisholm99]. A list of some of these tools can be found at the Web Accessibility Initiative (WAI) site [Abou-Zahra06].

The tools listed in the WAI site, either in the form of software programs or online services, aim to determine if a web site is accessible to people with disabilities. Some of these tools start to determine the conformance of Web sites to accessibility checks that can be executed automatically. Some also suggest checks that need to be evaluated manually.

Even though most of the listed tools are used to accessibility validation or verification, different tools identify different possible issues. Brajnik compares two of these tools and characterise them according to their completeness, their correctness and their specificity [Brajnik04]. One aspect relevant from this study is the high number of false positive and negative issues reported by these tools. One of the tools reported more than 80% false negative issues generated by automatic or manual tests.

3.2 DRUM

The Diagnostic Recorder for Usability Measurement (DRUM) [Macleod93] is a CAUSE tool from the National Physical Laboratory. It is a tool that supports quantitative evaluation based on observation of task performance and

analysis of how successfully people achieve task goals when using a system.

DRUM supports several steps associated with UE. It manages data acquired during the observational evaluation, allows the evaluator to select users and provides facilities for editing and browsing tasks during the analysis stage. DRUM is divided into four modules described next.

The **Evaluation Manager** allows the manipulation and display of data from the different stages of the UE. This manager has eight browser, one for each type of data: Subjects, Tasks, Recording plans, Evaluations, Video sessions, Logs, Measures and Reports.

The **Scheme Manager** enables evaluators to inspect existing events and add or edit custom events. Custom events may represent sub-tasks or activities to create a hierarchical description of the tasks to be performed. This hierarchy can have up to four levels. Once a task scheme is created, it is stored in the DRUM database.

The **Recording Logger** has two sub-modules that provide the functionality needed to create and edit video-related logs. The Recorder assists log creation in real-time, while the Logger assists "retrospective" logging. To log events of interest and easily locate them, evaluators can specify Markers. The Logger also controls the video recorder, manipulating the tape as needed. If an event is not instantaneous, DRUM offers the possibility to view the associated clip. For that the evaluator only needs to click on that event.

The **Log Processor** performs the calculation, from any log in the DRUM database. The performance measures and performance-based usability metrics include: task time, help and search time, effectiveness, efficiency, etc.

One advantage of DRUM is its ability to control a video recording device and search for meaningful events. According to the authors, the use of time-tamped observed events, may reduce the time required to analyse a video by up to 80% [Macleod93].

DRUM's dependency on specific hardware is an obstacle to the adoption or even further development of this tool.

3.3 KALDI

The Keyboard/Mouse Action Logger and Display Instrument (KALDI) is a CAUSE tool that supports UE of Java-based graphical interfaces [Ivory01]. It is more than an action logger, as it is able to capture usage data and screen shots for Java applications. This software solution eliminates the requirement for specialised video recorder.

KALDI has three components: the Capture Tool, the Monitoring Tool and the Analysis Tool. The first two components are used during the usability test, while the Analysis Tool serves to assist in the evaluation.

The **Capture Tool** logs user events and takes up screenshots using the Java AWT functionality. This collected data is associated to a time stamp and sent to the Monitoring Tool over a network connection.

The **Monitoring Tool** enables the UE expert to structure and store the information collected by the Capture Tool.

The **Analysis Tool** assists the study of recorded events and associated pictures after the execution of the usability test. The analyst may associate several “raw events” to Actions or Tasks, building a meaningful hierarchy. A player is also available to play back screenshots taken during the test.

3.4 UsAGE

The User Action Graphing Effort (UsAGE) [Uehling95] is a usability tool that compares two files: one “expert” with one “novice”. These files are generated by a utility called *usage_collect* of TAE Plus, which is a user interface development and management system.

After performing the comparison, UsAGE displays the result in a graph of action nodes. The “expert” series of actions is used as a reference and the “novice” actions are displayed as a comparison. In addition, it displays some metrics about the comparison results, including percentage of matches between expert and novice nodes.

3.5 Mercury TestDirector

The Mercury TestDirector [Mercury06] is a commercial application that supports the testing process: requirements management, planning, scheduling, running tests, issue management, and project status analysis. It uses a Web interface to allow testers and developers to participate and contribute to the testing process.

The testing process starts with the definition of the test requirements. Based on the requirements, testers build the test plan and design the actual tests. The next phase is to test the system as a whole. Tests can be scheduled and dependencies between them may be specified between them. Manual test execution is performed through a browser-based wizard that provides step-by-step guidance to the executer. Once this process is complete, TestDirector generates graphs and reports with the results of the tests.. It also provides an API that enables it to be integrated with custom solutions.

4. REQUIREMENTS

In a survey that presents a taxonomy for usability evaluation models only three from the ten methods used in user testing, have automation support: performance measurement, log file analysis, remote testing [Ivory01]. This may be an indicator of the difficulty to automate some steps in usability evaluation. Automated capture enables the evaluator to collect data for a larger number of users than traditional methods. In addition, automatic generation of usage data may reduce considerably the time required to review recorded testing sessions [Ivory01].

The UsAGE approach to data collection has two weak points: it requires the interface to be developed using TAE Plus and requires an expert execution of the task. The first point prevents it to be used with most commercial applications. The second weak point is the notion of “expert” versus “novice” that is not always applicable.

Even after the identification of issues, their presentation may not be consensual. Classification of issues is frequently a subjective activity. It may vary between evaluators or even for on evaluator in different conditions [Brajnik04]. Having customizable ranking in the evaluation software may clearly identify the priorities for problem solving. If circumstances change, a simple modification of the priorities will reorder the issues found in the UE.

The discussion of methods and systems presented above was used to identify the requirements for the proposed integrated usability evaluation environment. The identified requirements are:

- Platform independence and open source;
- Dissociable from the developing environment;
- Capable to record user actions;
- Able to record video and audio streams;
- Support for planning the user evaluation;
- Integrated counting and statistical engine;
- Ability to generate reports;
- Extensible and modular;

5. IMPLEMENTATION

One of the requirements for the proposed tool is to be dissociated from the software to be tested. We believe that the proposed solution has minimal intrusion relative to the user’s system.

As shown in Figure 1, our tool integrates three main components and two auxiliary ones, described in the remaining of this section.

The **Experiment Planner** enables the UE expert to develop a priori description of the tasks involved in the evaluation session.

The plan resulting from the Experiment Planner provides guidelines to run the experiment. The evaluator may define the Actions that define the user experiment.

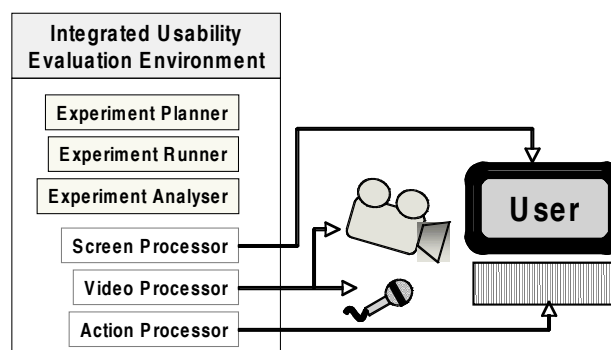


Figure 1: Overview of the I-UsE

The **Experiment Runner** controls the screen, video and action processors. In the beginning of each experiment, it instructs all the processors to start recording.

It is through the Runner that is identified the state of the current stage of the user experiment.

The **Experiment Analyser** enables the UE expert to analyse the recorded data.

The Analyser communicates with the processors to playback the corresponding stream. The analyst may introduce new marks or identify high-level Actions.

The **Screen Processor** enables the evaluator to capture and playback the user's screen. To achieve good screen resolution with the use of off the shelf components, we decided for a software solution.

To capture the user screen we use the free remote control software TightVNC. It is derived from VNC [Richardson98] software and has a Java implementation, making it easy to import as a system module. To capture the user's screen, a VNC server must be running on the user's computer. However, the VNC server does not send keystrokes nor mouse events. Thus, it was necessary to add the Action Processor.

The **Video Processor** manages video and audio streams. It may record live streams or playback existing ones.

To capture live video and audio using general purpose hardware we decided to use the Java Media Framework (JMF). Any video and audio hardware that is recognised by the JMF can be used to capture the live stream. Therefore, it is possible to use regular web cams and microphones as capture devices.

The **Action Processor** is a module in the planning phase. The goal is to manage user events: capture those events from the user machine and make them available to be combined with the remaining captured data.

6. CONCLUSION AND FUTURE WORK

In this paper we have presented the ongoing work concerning our tool designed to integrate data collection and UE analysis features. It was developed to tackle the problem of the lack of efficiency during this UE data study phase that leads to time consumption and consequently expensive studies.

In the future we plan to use our integrated usability evaluation environment in order to assist on the evaluation of a complex commercial system and therefore validate our claims of significant contribution to efficiency of the UE study phase.

We plan to extend our tool by introducing a domain specific (DSL) visual language to allow the usability expert to model UE sessions and generate tests automatically.

The technology used in this project enables to configure the tool for remote usage. This functionality enables a UE expert to guide the testing remotely, with users possibly being physical distant. The potential of this use case to our tool is a promising approach to reduce costs related to mobility and we plan to explore it further.

7. REFERÊNCIAS

- [Abou-Zahra06] S. Abou-Zahra, et al., Web Accessibility Evaluation Tools. W3C, March 2006.
www.w3.org/WAI/ER/tools/Overview.html
- [Brajnik04] G. Brajnik. Comparing accessibility evaluation tools: a method for tool effectiveness. *Universal Access in the Information Society*, Vol. 3, No. 3-4, Springer Verlag, Oct. 2004. pp. 252-263.
- [Chisholm99] W. Chisholm, G. Vanderheiden, I. Jacobs. *Web Content Accessibility Guidelines*. W3C Recommendation, May 1999.
[w3.org/TR/1999/WAI-WEBCONTENT-19990505](http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505)
- [Gediga02] G. Gediga, K. Hamborg, I. Düntsch. Evaluation of Software Systems. in A. Kent, J. Williams (Ed.), *Encyclopedia of Library and Information Science*, Vol. 72. pp. 166-192.
<http://citeseer.ist.psu.edu/gediga02evaluation.html>
- [Faulkner03] L. Faulkner. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, Psychonomic Society, Inc., 2003, Vol. 35 (3), pp. 379-383.
- [Hilbert00] D. Hilbert, D. Redmiles. Extracting Usability Information from User Interface Events. *ACM Computing Surveys*, Vol. 32, No. 4, December 2000, pp. 384-421.
- [Ivory01] M. Ivory, M. Hearst. The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys*, Vol. 33, No. 4, December 2001, pp. 470-516.
- [Macleod93] M. Macleod, R. Rengger. The Development of DRUM: A Software Tool for Video-assisted Usability Evaluation. *BCS Conference on People and Computers VIII, HCI'93*, Loughborough, 1993.
www.usabilitynet.org/papers/drum93.pdf
- [Mercury06] Mercury Interactive Corp. *TestDirector for Quality Center, Data Sheet*
<http://www.mercury.com/us/pdf/products/datasheets/DS-1134-0306-testdirector.pdf>
- [Molich04] R. Molich, M. Ede, K. Kaasgaard, B. Karyukin. Comparative usability evaluation. *Behaviour & Information Technology*, Taylor & Francis Ltd, Jan-Feb 2004, Vol. 23, No. 1, pp. 65-74.
- [Richardson98] T. Richardson, Q. Stafford-Fraser, K. Wood, A. Hopper. *Virtual Network Computing*. *IEEE Internet Computing*. Vol. 2, No. 1, 1998. pp. 33-38.
- [Uehling95] D. Uehling, K. Wolf. *User Action Graphing Effort (UsAGE)*. CHI 95 Short Papers, ACM, Colorado, USA. May 1995

Cognitive walkthroughs in the evaluation of user interfaces for children

Alexandre Mano José Creissac Campos
Departamento de Informática/CCTC
Universidade do Minho, Braga, Portugal
amano@bragatel.pt/josé.campos@di.uminho.pt

Abstract

This paper describes a case-study, dealing with the application of the cognitive walkthrough as a method of evaluating an interface built for children. We performed the walkthrough and tested the interface with children aged between 5 and 7 years old. Given our goals and the scope of this study, the cognitive walkthrough proved as a reliable source of indications about usability problems on an interface aimed at children.

Keywords

Child-computer interaction, usability, cognitive walkthrough,

1. INTRODUCTION

Every software project has to deal with an awkward situation: its designers are not the real, targeted users. That is why it is always mandatory to submit the project's interface to a rigorous scrutiny, which allows the finding of flaws and less-than-optimal features. To that effect, a number of analysis methods have been developed and used with considerable success [Desurvire92].

Software engineers usually resort to empirical usability methods to analyse their interfaces. When the project reaches a stage that allows testing, its behaviour is simulated for a group of human users. Starting with the observation of these users, along with interviews and inquiries, problems related to the interface are discovered and sorted out.

Testing children's interfaces though, is a different matter. Not only children are more reluctant to speak their opinions aloud, they are not "typical" users. Their needs and limitations make usability testing less straightforward than doing the same procedure with adults. Moreover, children are typically less experienced using computers and frequently are at a loss when describing their difficulties when experimenting with software.

This paper describes a case-study, dealing with the application of the cognitive walkthrough as a method of evaluation an interface built for children. The cognitive walkthrough seems like a perfect fit for children. After all, it aims to discover what can go wrong when the users have no previous knowledge of the system. In most cases, that's exactly the case with young children. They have little to no experience with the software they use at school or home, and frequently have to find their own way around it.

Oddly, to the authors knowledge, there are no published descriptions of studies that relate the findings of a cognitive walkthrough performed on an interface for children and the actual problems the children faced when using the software. This is the case here. We performed the walkthrough and tested the interface with children aged between 5 and 7 years old. Their results are compared and discussed.

2. THE COGNITIVE WALKTHROUGH METHOD

The cognitive walkthrough is an interface evaluation technique, which focuses on the support the interface offers to exploratory learning, i. e., usage with no previous training [Rieman95]. The method aims to verify how much can the system guide an untrained user, allowing the user to accomplish his goals.

This technique is based on Lewis and Polson CE+ theory of exploratory learning [Polson90]. The requirements for the application of the cognitive walkthrough are [Abowd95, Rieman95]:

1. A system description or prototype, detailed enough to allow for a complete navigation;
2. A set of representative tasks that an user can execute within the system;
3. The list of necessary actions to perform the tasks;
4. A description of the typical users, including their experience and expectations about the system.

After meeting these requirements, the cognitive walkthrough aims to simulate the steps taken by a user with no previous experience with the interface. During the execution of the tasks, the analyst registers the answers to the following questions [Abowd95]:

- 1) Will the users be trying to produce whatever effect the action has? – Ask whether an action appropriate to what the users would want to do at that point is available on the interface.
- 2) Will the users be able to notice that the correct action is available? – Ask whether the control that triggers the action is visible to the users when they need to use it.
- 3) Once users find the correct action at the interface, will they know that it is the right one for the effect they are trying to produce? – Ask whether the users will know that the control available is the one they need to complete the action.
- 4) After the action is taken, will users understand the feedback they get? – After the action is completed, ask whether the users will know the action is complete and what its effects are.

The answers provided by the walkthrough may lead to a set of flaws and possible improvements to the system interface. It is an asset that the analyst is familiar with the basic principles of interface design, as proposed by Lewis and Polson. These principles increase the chances that a user with no previous knowledge about the system will make correct choices through guessing.

3. BEFORE THE COGNITIVE WALKTHROUGH

The “História do dia” site (“Daily story”, <http://www.historiadodia.pt>) was chosen for the test. This site was chosen for its popularity among Portuguese children. Besides, it is targeted to the same age group as this study.

The method used to validate the cognitive walkthrough on interfaces for children was broken up into 5 steps:

1. Performing the cognitive walkthrough on all the tasks available on the web site. (performed by first author)
2. Creating a list of typical tasks that children perform on the web site.
3. Getting 2 other analysts to perform the cognitive walkthrough on the list created on the previous step.
4. Testing the tasks with children, and recording their results.
5. Comparing the difficulties experienced by the children and the problems predicted on steps 1 and 3.

Therefore, a “percentage of predictable problems” can be calculated on step 5, and this percentage can be compared to other tests with adults. That way, it is possible to compare the cognitive walkthrough performance when applied to children.

It is important to note that the usability problems faced by children were always sub-sets of the problems found on the analysis. If the children had found some problems which had not been predicted, the “percentage of predictable problems” would have to be adjusted.

4. EXECUTING THE COGNITIVE WALKTHROUGH

The tasks chosen for the test were: Read the daily story; Vote on the daily story; Hear the daily story; Read a story from the archive. The first 3 tasks are the most typical on everyday use of the site (according to a primary school teacher). The last is one of the other ten tasks available on the home page.

Users: Children aged 5 to 7. Some skill using the mouse and keyboard is assumed, as well as basic knowledge of web navigation. Users can read at an appropriate level for their age.






Figure 1. “História do dia” home page


Starting Point: “História do dia” home page (figure 1).

Tasks and actions

Task 1: Read the daily story.

- a. Click the picture associated with the story.
- b. Click the  icon to start the story.
- c. Click the  and  icons to navigate through the story.

Task 2: Hear the daily story.

- a. Click the picture associated with the story.
- b. Click the  icon to start the narration.

Task 3: Read a story from the archive.

- a. Click the “Arquivo” icon.
- b. Click the desired month.
- c. Click on the period within the chosen month, in which the story was first made available.
- d. Click the picture associated to the story.

Task 4: Vote on the daily story (includes task 1).

- a. Click the “Votar” icon.
- b. Choose how many stars will be awarded to the story.

As stated above, the first author went through the whole interface, and identified possible usability problems. However, the analysis of a single evaluator is typically not considered sufficient to uncover enough issues on any

given interface. Hence, a school psychologist, and a primary school teacher were asked to perform the walk-through on the same tasks the children would be confronted with. Both were skilled computer users, and experienced web users. Moreover, they worked with children on a daily basis, and therefore knew what could be expected from them when the interface came to test. As they had no previous experience on the evaluation of interfaces, some guidance was provided, using [Mano04]. Their results were recorded using the same kind of tables the first author used. One of these tables is shown below as a sample. A "Y" (Yes) on the table represents a well defined action. A "N" (No) identifies a possible usability problem on the interface.

Actions	Walkthrough questions			
	1)	2)	3)	4)
a)	Y	Y	Y	Y
b)	Y	N	N	Y
c)	Y	N	N	Y

Table 1. Execution of task 1 (read the daily story)

So, on this particular task, 2 usability problems were identified, on actions b) and c), indicating children may have trouble performing the task.

A full account of the answers given can be found on [Mano05] and the walkthrough results are summarized on table 7.

5. TESTS

The subjects were asked to perform the tasks in this order: read the daily story; vote on the daily story; read a story from last week; hear the daily story.

As the test was not supposed to last more than 15 minutes, the tasks were always presented in the same order. If the test was meant to take longer, randomizing the sequence would be advisable [Hanna97].

The tests were made on a primary school library, during class time. Before starting the test, the children were asked some questions, to establish some background on their experience and knowledge using computers and the web. This was a homogeneous group regarding computer usage. Every child had some skill using the keyboard and mouse, and they knew what a web page was and how to navigate. None of them had ever used the "História do Dia" web site. The test was performed individually, and their teachers were asked to keep their pupils busy before and after the tests, so they could not tell each other what to do.

The results were recorded during the tests. For each action, the following classification was used: action not completed; action completed; action completed after request for help; action completed after one or more failed attempts; action not executed, because a previous failure prevented the attempt. Tables 2 to 5 show the results.

	Action a	Action b	Action c
not completed	-	2	-
completed	7	6	9
with help	-	1	-
with failures	3	1	1

Table 2. Execution of task 1 (read the daily story)

	Action a	Action b
not completed	4	5
completed	6	2
with failures	-	1
previous failure	-	2

Table 3. Execution of task 2 (hear the daily story)

	Action a	Action b	Action c	Action d
not completed	10	-	4	2
completed	-	6	2	3
with failures	-	-	-	2
previous failure	-	4	4	3

Table 4. Execution of task 3 (read a story from archive)

	Action a	Action b
not completed	1	2
completed	8	5
with failures	1	2
previous failure	-	1

Table 5. Execution of task4 (vote on daily story)

6. CONCLUSIONS

It is necessary to establish criteria to consider whether the action is correctly defined on the interface. Considering this is a web site for young children, and the study's goals, it was decided that a success rate below 70% indicated the action should be presented some other way. Therefore, test results can be compared to the cognitive walkthrough predictions, and this is shown on table 7, which relates successes (S) and failures (F) for each action. When the action could not be tried due to a previous failure, it was not taken into action for calculation.

For example, when evaluating action b) on task 1, we found that 6 children out of 10 completed the task with no failures and no help, so we awarded this action a 60% success rate, which in our study is not enough to indicate a successfully presented action. So, on table 7, the action was given an "F" on the test results.

	Confirmed predictions	%
Author	8	73%
Psychologist	8	73%
Teacher	7	64%

Table 6. Predictions confirmed on tests

Table 6 shows how well the evaluators fared in terms of confirmed predictions. As 11 actions were tested, 8 correct predictions correspond to 73% accuracy. These prediction rates are higher than the ones obtained in a comparison between interface evaluation methods [Desurvire92], where they varied from 44% and 8%, using heuristic evaluation and 28% and 8% with the cognitive walkthrough. Obviously their testing was performed on more complex interfaces, but the results are encouraging.

Another good indicator that the cognitive walkthrough is a reliable evaluation method lies in the very fact that two inexperienced analyst's results came very close to those of the author himself, and, more importantly, to the results of the tests. This indicates that no great coaching or experience is necessary to get good results from the method.

As shown on table 7, the quality of analysis depends heavily on the analysts. While the author, psychologist and teacher reached similar percentages, only 4 tasks out of 11 had the same prediction. But the combined analysis identified every problem the children experienced. This confirms that using more than one analyst increases the method's reliability and allows for the discovery of a greater number of interface design problems. Besides, it doesn't require a large number of analysts to get a good analysis.

Given our goals and the scope of this study, the cognitive walkthrough proved as a reliable source of indications

about usability problems on an interface aimed at children. Further research will validate these conclusions, using various software types and different subjects.

7. REFERÊNCIAS

- [Abowd95] Abowd, G. Performing a Cognitive Walkthrough. Georgia Institute of Technology, College of Computing.
<http://wwwstatic.cc.gatech.edu/classes/cs3302/documents/cog.walk.html>
- [Desurvire92] Desurvire, H., Kondziela, J., and Atwood, M. What is Gained and Lost When Using Evaluation Methods Other Than Empirical Testing. In the proceedings of HCI 1992, Cambridge University Press, (York, UK, September 15-18, 1992), 89-102.
- [Hanna97] Hanna, L., Ridsen, K., and Alexander, K. Guidelines for Usability Testing with Children. *Interactions*, 4(5), (September + October, 1997). 9-14.
- [Mano04] Mano, A., Campos, J. C., *Aplicação de um Cognitive Walkthrough – estudo de caso*, Interacção 2004 – 1ª Conferência Nacional em Interacção Pessoa-Máquina, pp. 256-258, Grupo Português de Computação Gráfica, July 2004.
- [Mano05] Mano, A., *Interfaces de computador para crianças – avaliação e construção*. MSc. Thesis. Departamento de Informática, Universidade do Minho, September 2005.
- [Polson90] Polson, P.G. and Lewis, C.H. Theory-Based Design for Easily Learned Interfaces. *Human-Computer Interaction*, 5, 191-220, 1990.
- [Rieman95] Rieman, J., Franzke, M., and Redmiles, D. Usability Evaluation with the Cognitive Walkthrough. In CHI 95 Conference Companion (Denver, Colorado, May 7-11, 1995), 387-388.

Task	Action	Cognitive walkthrough prediction			Test results
		Author	Psychologist	Teacher	
1	a)	S	S	S	S
	b)	F	F	F	F
	c)	F	S	S	S
2	a)	F	S	S	F
	b)	F	F	F	F
3	a)	F	S	S	F
	b)	S	S	S	S
	c)	S	F	F	F
	d)	S	F	S	S
4	a)	F	S	F	S
	b)	F	F	S	F

Table 7. Comparison between cognitive walkthrough and test results

POSTERS E DEMOS

A System to Evaluate and Understand Routing Algorithms

Mario Marcelo Berón
Univ. de San Luis
Argentina
mberon@unsl.edu.ar

Pedro Rangel Henriques
Univ. de Minho
Braga
prh@di.uminho.pt

Maria Joao Varanda
Inst. Politécnico de Bragança
Bragança
mjoao@ipb.pt

Abstract

In this communication in the form of poster and demo we intent to present a tool aimed at evaluating and understanding routing algorithms. This application has a powerful environment that allows: to draw graphs and manipulate their components in manual or automatic form; to incorporate routing algorithms; to make experiments, measurement, and performance tests; to understand programs. The tool incorporates mechanisms to make easy the task for building executable files. These characteristics make this tool a versatile and easy evaluation and comprehension system to be used with routing algorithms.

Along the paper we will emphasize how we conceived and implemented the system's interface to satisfy the user's requirements. Our purpose in the context of Interaccao06 is to demonstrate it!.

Key words

Routing Algorithms, Graph Class, Metrics, Visualization, Comprehension Functions

1.INTRODUCTION

A graph is a tuple $G=(V,E)$ where V is a set of vertices and E is a binary relation defined on V . A graph is a very useful mathematical model to represent real systems (like networks, dependencies, etc.). There are many well known and optimized algorithms to implement the most usual operations over graphs. A routing algorithm is a program used to find one or more paths between two nodes of a graph.

Nowadays there are many routing algorithms tuned to solve each different case efficiently. For example, the Dijkstra's algorithm solves the problem of finding the shortest path between two nodes of a graph. However, the technological advances create new application contexts where traditional algorithms can't be applied; for example the mobile networks. In these applications the network's nodes change of place continuously and therefore the graph that represents the network is unpredictable, it changes dynamically. To solve this problem the scientific community created new routing strategies; for instance the online routing algorithms. The problem is that we don't know the performance of these new routing procedures and sometimes their operation and implementation is not clear.

To attack this problem we conceived a system to experiment, evaluate and understand routing algorithms. The evaluation is carried out through the execution of experiments programmed by the user. To aid the user in understanding the algorithms, the system provides several views of the program under analysis, from the control flow to the machine code.

This communication is organized as follows. Section 2 describes the functional requirements of the routing system. Section 3 shows the approach used to visualize graphs and paths. Section 4 explains the mechanism of automatic compilation and the interaction with other tools. Finally, section 5 presents the comprehension facilities.

2.REQUIREMENTS

In this section we describe the requirements expected from a good system for evaluation/comprehension of routing algorithms. The selection of the criteria is based on the study of the state of the art and on the experience of our research group.

A routing evaluation/comprehension system must provide the following main functions:

1. Visualization of the graph and the path found by each routing algorithm.
2. Incorporation of different routing algorithms, support for graph classes, metrics and experiments.
3. Embedded compilation, linkage and loading processes.
4. Interaction with other tools, such as compilers, spread sheets, text editors, etc.
5. Algorithm Comprehension processes.

For the implementation of these requirements, we built a graphical environment composed by a menu with tool bars and windows that integrate all theses function.

In the following sections we justify these needs and we describe our solution.

3.VISUALIZATION OF GRAPHS AND PATHS

The visualization of the different kind of graphs and paths is necessary because the user needs to analyze the network type used and to study the routing strategies.

To satisfy this requirement, the evaluator has a window that shows the graph used before having applied the routing strategies (see figure 1).

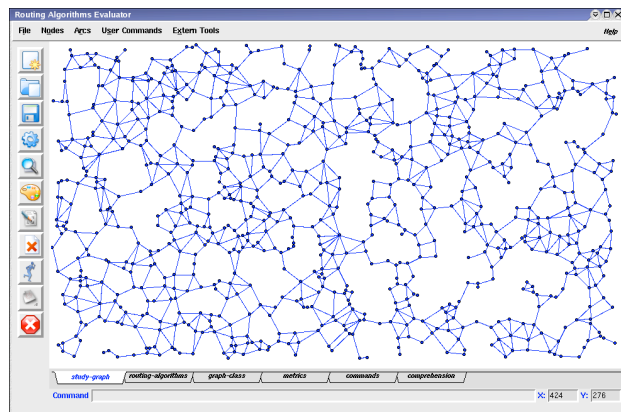


Figure 1: Graph visualization

Additionally, the user can see the path found by the routing algorithm chosen. The result can be visualized in two forms: i) as a trace; or ii) as a whole. The first one allows the user to decide step by step when to show the next node in the path. This task is made by using a dialog box (see figure 2).

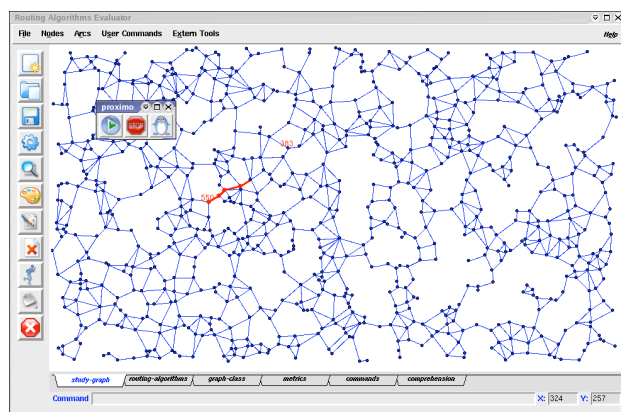


Figure 2: Path visualization using trace method

With the second approach it is possible to see the path without intervention of the user (see figure 3). Furthermore, the evaluator provides functions to: delete; personalize the color; identify and search nodes and arcs. These tasks are made in a simple form using the menu and tool bars.

Inside the graph visualization window, the user can create a graph manually, just clicking with the mouse on the drawing area. When the user wants to create arcs, he must make a click on the determined area for the source node. Then, without liberating the button, the

mouse's pointer must be placed on the determined area, for target node. Finally the mouse's button can be released.

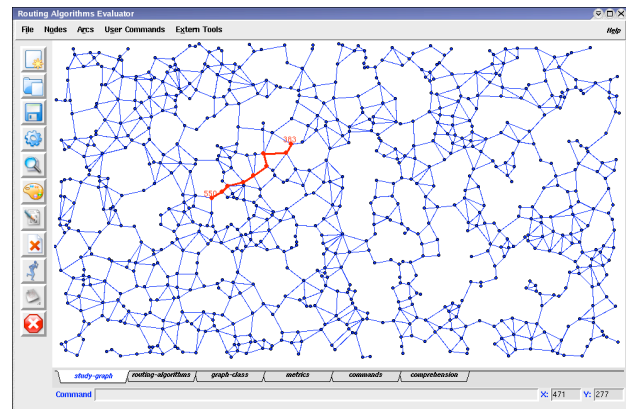


Figure 3: Walk visualization total

3.1 Incorporation of Routing Algorithms, Graph Classes, Metrics and Experiments

In the set of mandatory characteristics that an evaluation/comprehension system must have for routing algorithms, we include operations to incorporate the routing algorithms, graph classes, metrics and experiments easily. In this way, the user is able to focus on the study of routing algorithm study.

To satisfy these requisites, the routing system has four windows to make available each one of these components. Each window has a set of parameter that have to be defined.

For the routing algorithm case the parameters are: routing algorithm's name; function's name which implements it; and the function parameters. After having satisfied the parameter values, the user, presses the *apply* button and the system incorporates this routine to its kernel. Finally, he must press the *build object* button to build the object module of each routing algorithm.

Note that the routing algorithms can be eliminated using the button *delete*. Figure 4 shows the window for the routing algorithm incorporation.



Figure 4: Routing Algorithm window

The other three windows have a similar behavior. The experiment's window has an additional function that allows the automatic creation of dialog boxes. This characteristic has the objective of facilitating the tool usage by the experts. The commands can be executed from the command line or the dialog boxes.

4. AUTOMATIC COMPILATION AND INTERACTION WITH OTHER TOOLS

One of the main difficulties found in systems like this, is the complexity of the routines compilation process because the system has many routines in its kernel. These routines implement different data structures useful for the creation of new routing algorithms, graph classes, etc. In addition to this, the possibility that the user has to increment the system functions (remember that the user can incorporate new routing algorithms, graphs, metrics, etc.) produces an increase in the system's routines. For this reason, the system has a compilation mechanism based in the creation of object modules of each routine that then they are linked to build the executable file. All these tasks are easily made by pressing the button *build object* in the case of routing algorithms, graph classes and metrics. To experiment, the user just has to press the *executable* button. In this way, he is only concerned with the data parameters, and the system makes the rest.

Interaction with other tools

When the user studies the routing algorithms, he needs: i) to program the routing strategies and ii) to visualize the result. To make these tasks easier the system interacts with other tools such as: compilers, text editors, spread sheets. To access to each tool, the user only needs to activate the option *external tools* of the menu bar and then select the adequate application.

5.COMPREHENSION OF ROUTING ALGORITHMS

Sometimes the programmer has access to routines which implement a task like graph classes, routing algorithms, metrics or experiments that he needs to include in a certain application. However, usually he prefers to develop the program from scratch instead of reusing the one available because reusing implies to understand the code written by another person and that is usually a very hard task. To help in this task, our system has a set of functions aimed at program understanding.

The program comprehension can be reached building different views of the algorithms under study.

We thought that the following views are important: i) the result, that is to say, the output provided by the program; ii) the functions used to get the result and iii) the source and object code associated to each function.

The result is important because it allows the user to have an idea of the algorithm's objective. The list of function calls offer information about how the result was obtained. The two remaining views let that the function's implementation to be visualized and studied. The reader can see the comprehension screen with its four windows in figure 5.

In addition, the system provides navigation functions. Once obtained the list of function calls used to build the result, the user can observe its source and object code just clicking in the name of each function; the system reacts showing the code of the selected function automatically.

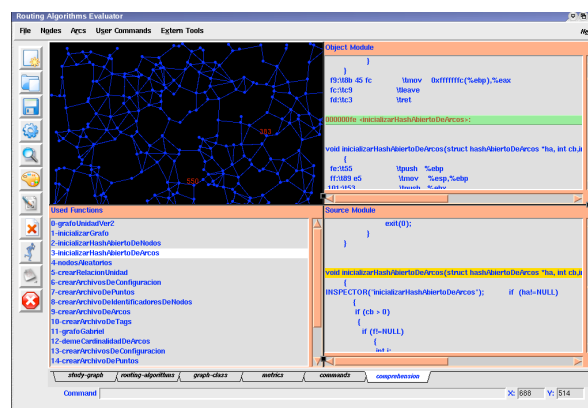


Figure 5: Comprehension Windows

On the other hand, the system permits the user to document each code (source and object) doing click on the corresponding line.

6.REFERENCES

[Berón05] Berón, M; Hernández Peñalver, G; Gagliardi, O. “Un Evaluador de Algoritmos de Ruteo”. Master Thesis in Software Engineering. Universidad Nacional de San Luís. 2005

[Brooks78] Brooks, R. “Using Behavioral Theory of Program Comprehension in Software Engineering”. IEEE. 1978.

[Riesco02] Riesco, D; Grosso, A; Berón, M. “Una Herramienta para la Ingeniería Inversa de Sistemas Escritos en Lenguaje C, Bajo Linux”. Comunicación, Expuesta y Publicada en el Acta de Resúmenes del Congreso Argentino de Ciencias de la Computación CACIC en el año 2002.

[Linter03] Linter, R; Michand, J; Storey, M; Wu, X. “Plugging-in Visualization: Experiences Integrating a Visualization Tool with Eclipse”. ACM Symposium on Software Visualizaton. 2003.

[Mayrhauser95] Mayrhauser, A; Vans, M. “Program Comprensión During Software Maintenace and Evolution”. IEEE. 1995.

- [Moreno04] Moreno, A; Myller, N; Satinen, E; Ben-Ari, M. "Visualizing Programs with Jeliot 3". ACM. 2004.
- [Oliveira06] Oliveira, E; Henriques, P; Varanda, M. "Características de um Sistema de Visualização para Compreensão de Aplicações Web através de Inspeção de Software e baseadas em Modelos Cognitivos". Tesis de Maestría en Ingeniería del Software. 23 de Febrero de 2006.
- [Pacione03] Pacione, M; Roper, M; Wood, M. "A Comparative Evaluation of Dynamic Visualization Tools". Proceedings of the 10th Working Conference on Reverse Engineering (WCRE'03). IEEE. 2003.

Conforto térmico humano: uma interface gráfica

Celina P. Leão
Dep. Produção e Sistemas, UM
4710-057 Braga
cpl@dps.uminho.pt

Pedro Arezes
Dep. Produção e Sistemas, UM
4800-058 Guimarães
parezes@dps.uminho.pt

S. F. C. F. Teixeira
Dep. Produção e Sistemas, UM
4800-058 Guimarães
st@dps.uminho.pt

Sumário

O conforto humano tornou-se num dos maiores desafios no desenvolvimento de novas tecnologias porque a eficiência/desempenho é grandemente influenciada pelo conforto térmico. Nesta categoria deve-se incluir o conforto nos locais de trabalho, tecnologias de transporte e automóveis, vestuário activo e vestuário de protecção. Foi desenvolvida uma aplicação informática, que integra um modelo matemático de termoregulação, uma base de dados com as constantes do modelo e ainda uma interface gráfica. A interface gráfica permite facilitar o uso do modelo de conforto térmico como uma ferramenta de aprendizagem não só a nível de graduação como também no ensino de pós-graduação. Foi feita uma avaliação da sua usabilidade no sentido de analisar a satisfação e a atitude do utilizador. Foram identificados alguns problemas os quais foram corrigidos. Posteriormente, uma nova avaliação foi elaborada. Neste momento a aplicação desenvolvida, ConfTermal, está a ser traduzida para inglês de forma a abranger maior número de interessados.

Palavras-chave

Conforto térmico, Usabilidade, Análise estatística, Ferramentas educacionais

1. INTRODUÇÃO

A análise do conforto térmico é normalmente efectuada recorrendo-se a alguns índices, que mais não são do que indicadores quantitativos desse conforto. Estes índices são, regra geral, calculados através de métodos mais ou menos complexos, mas que têm sempre por base os parâmetros climáticos relativos ao ambiente térmico em análise, bem como outros parâmetros individuais (Metabolismo, Isolamento do vestuário, Idade, Estatura, etc.).

A leccionação de algumas matérias, tais como esta questão do conforto térmico, poderá ser bastante melhorada se os alunos conseguirem perceber, com clareza, o papel desempenhado por alguns dos parâmetros mais relevantes para o problema em questão.

Por outro lado, a implementação do cálculo do conforto térmico em ambientes num software educacional representa um desafio interessante e complexo sob vários pontos de vista, em particular no que se refere à necessidade de cumprir os requisitos de utilização simplificada do mesmo.

Em termos computacionais, o desenvolvimento de uma interface gráfica, bem como a definição do menu de opções e as etiquetas a estes associadas não é uma tarefa simples e linear. O desenvolvimento e construção de uma interface gráfica deverá seguir alguns requisitos ergonómicos básicos. A usabilidade na interface apresentada ao utilizador é uma qualidade intrínseca ao próprio software. Tal deverá permitir que o utilizador

possa “navegar” através do programa informático com acções que correspondam às suas expectativas e, portanto, seja efectuada de forma eficiente [Babo96]. Uma interface pode, no entanto, ser funcional e bem concebida tecnicamente e ser, simultaneamente, pobre sob o ponto de vista da usabilidade. Tal acontecerá, sobretudo, se o software for desenvolvido sem ter a preocupação de conhecer melhor as necessidades específicas dos seus potenciais utilizadores.

A definição de condições relativas ao Conforto Térmico Humano (CTH) e a dispositivos associados a estes, como sejam o ar condicionado, a refrigeração, sistemas de aquecimento, assim como a análise ergonómica de postos de trabalho são importantes tarefas atribuídas aos Engenheiros Mecânicos. O CTH tem vindo a ser reconhecido como um dos parâmetros importantes no desenvolvimento de novas tecnologias, dado que a performance humana está fortemente relacionada com o CTH. Citam-se os exemplos da concepção de postos de trabalho, de tecnologias automóveis e de transporte, vestuário desportivo e de protecção.

A percepção do conforto térmico é principalmente afectada pela temperatura e pelos processos de transferência de massa entre o corpo humano e o ambiente em seu redor. De forma a facilitar a aprendizagem destas matérias e a ilustrar os efeitos associados a determinadas alterações, foi desenvolvido um software de análise do conforto térmico. O modelo desenvolvido efectua o cálculo das temperaturas de todo

o corpo humano ao longo de um dado período de tempo [Ferreira05].

2. APRESENTAÇÃO DA INTERFACE

O modelo de termo-regulação do sistema corpo-vestuário-ambiente utilizado neste trabalho é uma adaptação de um modelo já existente e publicado [Epifânio02]. O modelo térmico é definido por um conjunto de equações diferenciais ordinárias incluindo o processo de transferência de calor. O corpo humano é dividido em 16 partes distintas. Cada parte é dividida em 3 camadas: *core*, *shell* e pele. O coração e os pulmões constituem um sistema isolado e tratado em separado. No que diz respeito ao vestuário, várias camadas poderão ser previstas. Para o modelo mais simples, no qual apenas são consideradas 2 camadas de vestuário e as mãos e cabeça destapadas, este é definido por um conjunto de 70 equações diferenciais ordinárias.

As soluções numéricas são obtidas utilizando o método de Runge-Kutta-Merson devido à sua simplicidade e robustez [NAG06]. As soluções finais correspondem às temperaturas da pele em diversas partes do corpo, para diferentes tipos de isolamento do vestuário e diversos ambientes, bem como as temperaturas para as restantes camadas consideradas (Figura 1). Dado o elevado número de parâmetros, físicos e numéricos, foi desenvolvida uma base de dados que permitisse armazenar toda a informação de entrada assim como os resultados finais obtidos.



Figure 1. Interface de apresentação das temperaturas em todo o corpo e em todas as camadas, no início (esquerda) e no final (direita) da simulação.

O interface gráfico ConfTermal, compreende a programação numérica e as base de dados, foi desenvolvido em Pascal (Delphi 6) para o ambiente Windows. O utilizador, depois de executar a aplicação deverá utilizar sucessivamente vários menus. De acordo com a figura 2, as opções dos menus são: (1) introdução dos parâmetros iniciais, (2) efectuar a simulação numérica e a visualização dos resultados, (3) ver a base de dados, (4) cálculo do índice PMV, (5) menu de ajuda, (6) e opção para sair da aplicação.

De forma a ir de encontro dos requisitos educacionais associados à computação dos índices de conforto térmico, o cálculo do índice PMV (*Predicted Mean Vote*) foi incluído nesta aplicação. Os requisitos referidos foram confirmados num estudo anterior acerca desta aplicação [Teixeira05].

A opção relativa ao índice PMV permite o cálculo deste parâmetro através da introdução dos parâmetros de entrada necessários, nomeadamente o Metabolismo (M), o Isolamento do vestuário (Icl), Temperaturas (t_a , t_r), Velocidade do ar (v_a) e Pressão parcial do vapor de água (P_a). A equação utilizada para o cálculo do PMV corresponde a um modelo estacionário. Trata-se de uma equação empírica aplicada com o objectivo de prever o voto previsível médio, numa escala ordinal, da avaliação do conforto térmico de uma população significativamente grande.

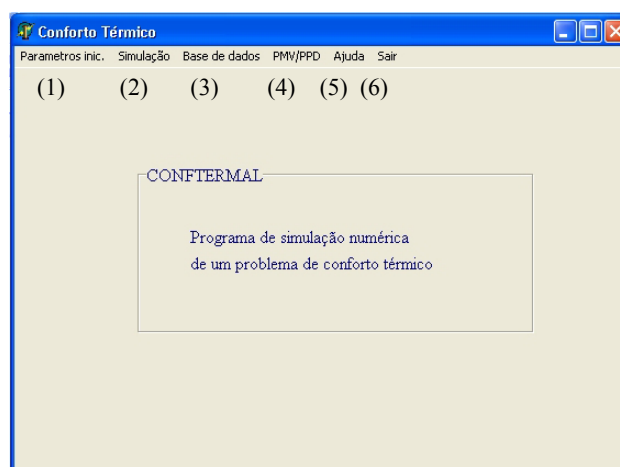


Figura 2. Menu da aplicação ConfTermal.

3. TRABALHO DESENVOLVIDO

O objectivo principal da aplicação ConfTermal é facilitar o uso do modelo matemático que descreve o conforto térmico. A avaliação de usabilidade da aplicação serve como uma ferramenta de extrema importância na análise da reacção dos prováveis utilizadores. A avaliação da usabilidade foi efectuada recorrendo-se à utilização e aplicação de 2 questionários previamente desenvolvidos e validados para o efeito. Com este método, não só se obtém as respostas de forma relativamente rápida e pouco dispendiosa, como também permite analisar a satisfação/opinião do utilizador final. Nesta avaliação foram utilizadas as versões portuguesas dos dois questionários utilizados frequentemente para medir a satisfação do utilizador SUMI (*Software Usability Measurement Inventory*) [SUMI06] e QUIS (*Questionnaire for User Interface Satisfaction*) [Chin88]. O questionário SUMI é um questionário extenso com 50 questões com três níveis de resposta: Concordo, Indeciso e Não Concordo. Através deste questionário, são testados cinco aspectos principais: Eficiência, Empatia (apreciação subjectiva), Controlo, Ajuda e Aprendizagem. O segundo questionário, o questionário QUIS, consiste em 27 questões, divididas em cinco

dimensões: “Reacções ao *Software*” (6 questões), “Ecrã” (4 questões), “Terminologia e Sistemas de Informação” (6 questões), “Aprendizagem” (6 questões) e “Capacidades do Sistema” (5 questões). Cada questão tem uma escala de avaliação de 10 pontos, que varia entre 0 (pior caso) e 9 (melhor caso). Os dois questionários foram aplicados em diferentes grupos de utilizadores pelo seu grau académico: Gestão Industrial (alunos dos 3º e 4º anos de Engenharia e Gestão Industrial da Universidade), Mecânica (estudantes do 4º ano de Engenharia Mecânica da universidade), pós-graduação (estudantes do curso de Mestrado e pós-graduação em Engenharia Humana da Universidade), Secundário (alunos do 12º ano do ensino secundário com conhecimentos na área de tecnologias da informação) e Professores (inclui vários professores da Universidade).

Nesta fase, os resultados obtidos sugerem que a interface gráfica foi bem aceite por todos os utilizadores dos vários grupos considerados. Foi possível observar que quando maior for o conhecimento na área de conforto térmico, mais críticos e exigentes se tornam os utilizadores no manuseamento da interface gráfica.

Um dos objectivos do desenvolvimento desta aplicação é o seu uso como uma ferramenta de aprendizagem para os alunos de graduação e de pós-graduação referente ao assunto do conforto térmico. Assim, a opinião dos utilizadores inquiridos referente aos tópicos de aprendizagem forma muito importantes. Os resultados finais foram motivadores pois, indicaram uma opinião positiva em todas as perguntas de aprendizagem.

Os aspectos menos positivos identificados disseram respeito à utilização da ajuda e documentação nela disponível e, sobre a representação gráfica final do corpo humano com as respectivas temperaturas. Esta última questão surge devido à enorme quantidade de informação que deve ser visualizada: temperaturas sobre o corpo humano nas 16 partes diferentes e para cada uma delas três camadas diferentes.

Tentando resolver este problema, foi permitido ao utilizador uma opção de só visualizar as várias temperaturas na camada escolhida.

4. TRABALHO FUTURO

Neste momento, a interface ConfTermal desenvolvida está a ser implementada em inglês alargando assim o universo de possíveis utilizadores.

Para uma melhor divulgação deste trabalho, está previsto a adaptação desta aplicação em ambiente *web*.

5. AGRADECIMENTOS

Nossos agradecimentos a POCTI/EME/62786/2004 pela ajuda financeiro e a todos os estudantes pela sua colaboração voluntária.

6. REFERÊNCIAS

- [Babo96] R.M. Babo, 1996, “A Avaliação da Usabilidade de um Sistema”, M.Sc.Thesis, University of Minho (in Portuguese).
- [Chin88] J. Chin, V. Diehl, K. Norman, 1988, “Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface”, Proceedings of Computer-Human Interaction, CHI Conference, pp. 213-218.
<http://www.lap.umd.edu/QUIS/>
- [Ferreira05] M.L. Ferreira, 2005, “Aplicação de um Modelo à Determinação de Índices de Conforto Térmico”, M.Sc. Thesis, University of Minho (in portuguese).
- [Epifânio02] P. Epifânio, A. Silva, S.F.C.F. Teixeira, and J.C.F. Teixeira, 2002, “Modelo de Conforto Térmico Baseado na Distribuição de Temperatura do Corpo Humano”, Proc. Métodos Numéricos en Ingeniería V, SEMNI, Spain (CD-ROM) (in portuguese).
- [NAG06] NAG – Numerical Algorithms Group webpage, <http://www.nag.co.uk/>
- [SUMI06] SUMI - Software Usability Measurement Inventory webpage,
<http://www.ucc.ie/hfrg/questionnaires/sumi>
- [Teixeira05] S.F.C.F. Teixeira, M.L. Ferreira, and L.G. Costa, 2005, “A Computer Interface for Human Comfort Calculations”, Proc. of IMC22, Ireland, pp. 131-137.

Desenho de um Sistema de Recolha de Estatísticas para o Boccia

Luís Alexandre Gonçalo Fontes Nelson Cavaco Luís Garcia
LabSI/ESTIG-IPBeja
Rua Afonso III, 1, 7800-050 BEJA

lm.alexandre@gmail.com
gonçalo.fontes@sapo.pt
nelson_cavaco@hotmail.com
luisbgarcia@estig.ipbeja.pt

Sumário

Neste artigo apresenta-se o desenvolvimento de uma aplicação para o registo em tempo real de estatísticas de jogo para o Boccia, um desporto praticado essencialmente por pessoas com necessidades especiais. O recurso a técnicas de desenvolvimento de sistemas interactivos como a análise de tarefas, a prototipagem e a continua avaliação das propostas de desenho, quer através do recurso a regras e princípios de usabilidade, quer através do feedback dos utilizadores revelou-se extremamente importante na concepção do sistema. Com este artigo os autores pretendem partilhar a sua experiência na aplicação destas técnicas assim como na utilização de plataformas computacionais móveis no desenvolvimento destes sistemas interactivos.

Palavras-chave

Desenho de Sistemas Interactivos, Análise de Tarefas, Prototipagem, Ambientes Móveis, Pessoas com Necessidades Especiais, Desenho Centrado no Utilizador.

1. INTRODUÇÃO

O Boccia é um desporto desenvolvido especialmente para pessoas com necessidades especiais. No Boccia cada jogador deve lançar um conjunto de bolas que devem ficar tão próximas quanto possível de uma bola alvo. A pontuação da jogada é atribuída em função da proximidade destas bolas em relação à bola alvo. Cada jogo é disputado por dois jogadores (ou duas equipas) lançando cada um 6 bolas. Um dos jogadores utiliza as bolas azuis, o outro, as bolas vermelhas (Figura 1).

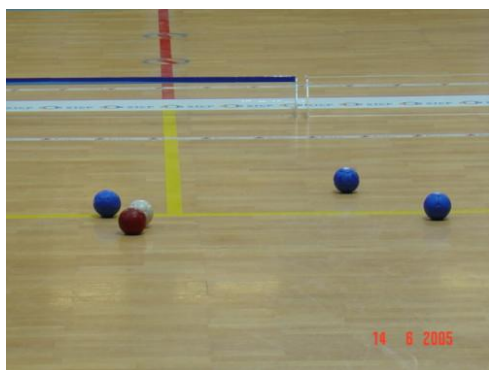


Figura 1 – Jogo de Boccia.

Um dos jogadores inicia o jogo lançando a bola alvo (bola branca). De seguida o outro jogador lança a sua primeira bola. O primeiro jogador volta a jogar e tem de colocar a sua bola mais próxima do alvo do que a bola do adversário. Enquanto não colocar uma bola mais junto do alvo tem de continuar até esgotar as bolas. Ao conseguir colocar uma das bolas mais próxima do alvo pára de jogar e o processo repete-se para o outro jogador. O jogo termina quando os jogadores esgotarem as suas bolas e ganha o jogador que tiver a bola mais próxima da bola alvo. Se a segunda bola mais próxima pertencer ao outro jogador o resultado é apenas 1-0. No caso de as bolas mais próximas pertencerem a um mesmo jogador então o resultado do jogo pode aumentar para 2-0, 3-0, etc. Uma partida é composta por 4 jogos, existindo a possibilidade de disputa de um 5º jogo em caso de empate [CP-ISRA05].

2. O PROBLEMA

Para a análise dos pontos fortes e pontos fracos de cada jogador a equipa técnica do Clube de Boccia do Centro de Paralisia Cerebral de Beja (CPCB) começou a classificar e registar os lançamentos dos vários jogadores em fichas de registo. Estes registos em papel permitem

esquerda longa	Centro longa	direita longa
esquerda média	Centro média	direita média
esquerda curta	Centro curta	esquerda direita
<i>Zona de Lançamento</i>		

Figura 2 – Divisão do campo de Boccia

depois uma análise do desempenho dos vários jogadores nas diversas situações de jogo.

Para o registo de um lançamento no formulário o técnico anota na zona associada ao jogador a posição da bola alvo (esquerda curta, centro curta, direita curta, esquerda média, centro média, direita média, esquerda longa, centro longa, direita longa). Estas áreas encontram-se representadas na Figura 2. Em cada lançamento o técnico regista a posição da bola branca, de seguida o tipo de lançamento: directo, bloqueio ou tabela, e por fim atribui uma classificação à jogada segundo os critérios: orientação e força. O critério orientação pode assumir três valores diferentes: (1) orientação correcta; (2) desvio para a esquerda e (3) desvio para a direita. O critério força também pode assumir três valores diferentes: (1) força correcta; (2) pouca força e (3) muita força.

Apesar de útil, o processo de registo dos dados num formulário em papel tem-se revelado moroso e permissivo a erros. Em fases posteriores a equipa técnica também sentiu algumas dificuldades na análise dos dados registados nos formulários.

Através deste projecto que ainda se encontra em curso pretende-se facilitar o registo e a análise dos lançamentos de cada jogador através do desenvolvimento de um sistema interactivo adequado a estas tarefas. Neste artigo os autores pretendem partilhar a sua experiência na aplicação de técnicas de interacção pessoa-computador assim como na utilização de plataformas computacionais móveis no desenvolvimento deste sistema interactivo.

3. DESENVOLVIMENTO DO SISTEMA

No desenvolvimento do sistema foi declaradamente assumida a máxima que indica que *“nada sai bem à primeira”* [Dix03]. Numa primeira fase foi efectuada uma análise de requisitos e em seguida utilizada de forma intensiva a prototipagem nas fases de desenho e implementação. Nestas fases foram desenvolvidos protótipos que eram avaliados e melhorados de forma iterativa. Na Figura 3 apresenta-se o percurso efectuado pela equipa

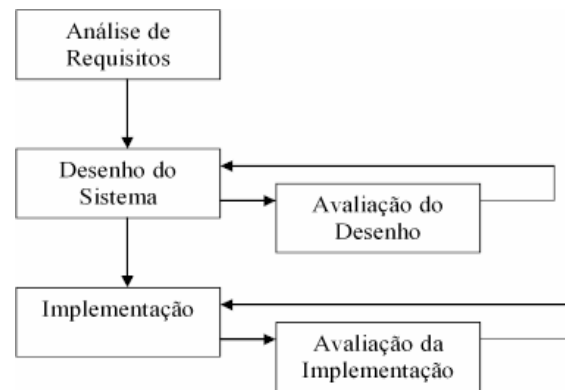


Figura 3 – Etapas do desenvolvimento do sistema

de desenvolvimento do sistema. Nas secções seguintes discutiremos as várias técnicas aplicadas nas fases de desenvolvimento.

3.1 Análise de Requisitos

Antes de iniciarmos esta fase efectuámos uma pesquisa bibliográfica sobre sistemas semelhantes. Nas várias fontes pesquisadas não encontramos qualquer referência a sistemas específicos para o Boccia. Podemos por isso dizer com algum grau de certeza, que este é um sistema pioneiro na área do Boccia. Estendemos a pesquisa a outros desportos e encontramos alguns sistemas comerciais noutras áreas, nomeadamente na área do futebol [Soccer05].

A análise de requisitos iniciou-se com um primeiro contacto com a equipa de Boccia do CPCB onde foram determinadas quais as expectativas dos futuros utilizadores em relação ao sistema. Como resultado foi elaborada uma lista de necessidades às quais o sistema teria de corresponder, nomeadamente ao nível do registo das jogadas e geração de estatísticas para consulta posterior. Após este primeiro contacto iniciou-se uma análise das tarefas realizadas pelo técnico no processo de registo dos lançamentos.

3.1.1 Análise de Tarefas

Esta análise de tarefas permitiu aos elementos da equipa de desenvolvimento o estudo das várias acções realizadas pelo treinador da equipa de Boccia. Para a realização desta análise foi consultada a equipa técnica de Boccia e estudados os registos de jogadas em suporte de papel. Esta análise permitiu um conhecimento sobre os vários termos técnicos utilizados na área (linguagem dos utilizadores) e o esclarecimento de dúvidas sobre as tarefas, quer com os utilizadores do sistema, quer entre os próprios elementos da equipa de desenvolvimento. Com base nas tarefas realizadas actualmente pelo treinador da equipa de Boccia foi analisada a forma como estas tarefas seriam realizadas no sistema informático. Na Figura 4

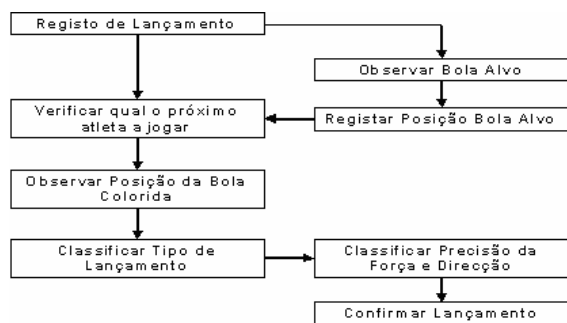


Figura 4 – Análise do registo de um lançamento

pode-se observar a análise da tarefa que consiste no registo de um lançamento no novo sistema.

3.2 Desenho do Sistema

Com base na informação obtida nas fases anteriores foram desenvolvidos protótipos de baixa fidelidade que depois de validados junto dos utilizadores permitiram a construção de protótipos de alta fidelidade que continuaram a evoluir em termos da sua usabilidade.

3.2.1 Protótipos de Baixa Fidelidade

Os protótipos de baixa fidelidade foram desenvolvidos com base na análise de tarefas realizada. A Figura 5 apresenta um dos protótipos de baixa fidelidade do ecrã principal da aplicação. A análise de tarefas direccionou em grande medida a disposição dos elementos na interface. Neste caso particular decidiu-se dividir o ecrã em duas partes, a parte mais à esquerda para a indicação do posicionamento da bola branca assim como do jogador que realizou o lançamento, e a parte mais à direita para a classificação da jogada. A adopção desta disposição favorece uma execução sequencial (em forma de U invertido) e cíclica da tarefa do registo de um lançamento. Neste protótipo também é visível a aplicação de certos princípios e regras de usabilidade. Por exemplo no final do registo de um lançamento o utilizador recebe *feedback* indicando a conclusão da tarefa, que pode ser anulada caso não seja iniciado um novo registo. Somente nesta estrutura simples encontram-se incorporados quatro das oito regras de usabilidade apontadas por Shneiderman: (1) *oferecer feedback informativo*, (2) *desenhar diálogos com princípio, meio e fim*, (3) *oferecer prevenção e recuperação de erros* e (4) *permitir desfazer facilmente as operações* [Shneiderman04]. Um outro exemplo consiste na utilização de um campo de Boccia para a indicação da posição da bola branca numa das nove zonas possíveis favorecendo assim a ligação entre o sistema e o mundo real, uma das dez heurísticas de Nielsen [Nielsen00]. O mesmo princípio foi aplicado para a indicação da força e precisão do lançamento. Cada uma das setas ao redor da bola com o sinal + (lançamento perfeito), indica o local para onde foi efectuado o lançamento, neste caso defi-

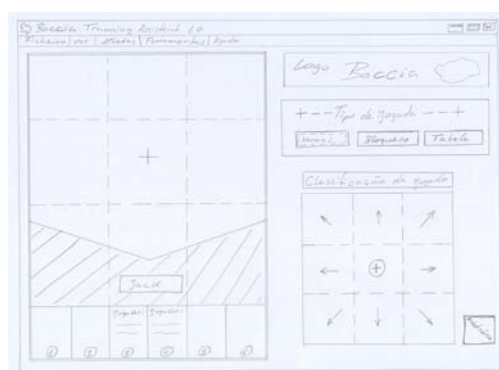


Figura 5 – Protótipo de Baixa Fidelidade do Sistema

ciente. Por exemplo a seta do canto superior esquerdo indica que o lançamento foi efectuado com força a mais e desviada para a esquerda. Através desta ligação com o mundo real pretendeu-se facilitar a aprendizagem e desempenho do utilizador com o sistema.

3.2.2 Avaliação do protótipo de baixa fidelidade

Como referimos anteriormente o processo de desenho dos protótipos de baixa fidelidade esteve sempre em constante avaliação através do confronto das várias opções com diversos princípios e regras de usabilidade. Após a obtenção destes primeiros protótipos procedeu-se à sua avaliação junto dos utilizadores. Nesta avaliação os utilizadores foram confrontados com as interfaces e folhetos explicando o seu modo de funcionamento. As interfaces foram consideradas extremamente simples e sugeridas algumas alterações. Uma primeira sugestão relacionou-se com a forma de escolha dos jogadores intervenientes num jogo. O protótipo inicial (Figura 6) previa que a escolha dos jogadores intervenientes, fosse efectuada por intermédio de seis caixas de selecção do tipo *drop down*, estando as mesmas dispostas em duas colunas. Cada uma destas colunas continha três caixas *drop down*, de forma a poder representar uma equipa de boccia, caso a situação de jogo fosse outra (pares ou individual), a 2ª e ou 3ª caixas ficariam desactivadas. Após a sugestão apresentada pelos utilizadores, esta opção sofreu uma remodelação, passando a escolha dos jogadores a ser efectuada, através de uma representação do

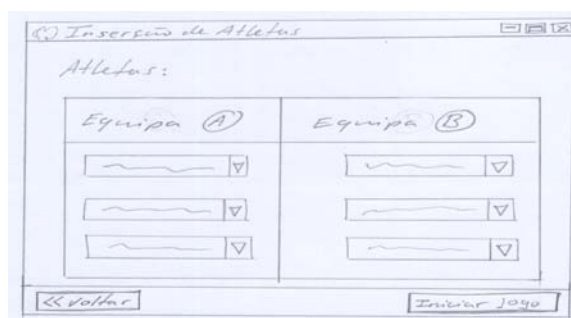


Figura 6 – Protótipo inicial da escolha de jogadores

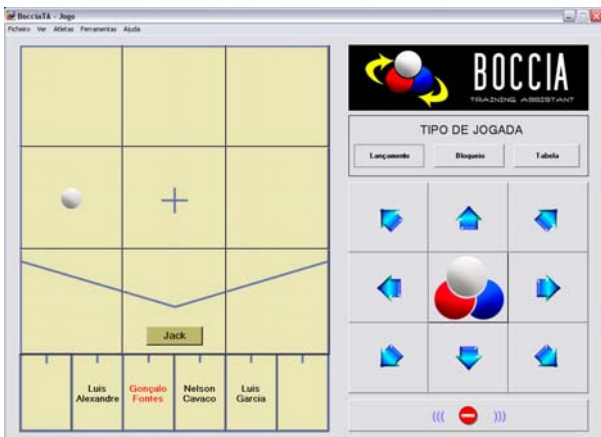


Figura 7 – Ecrã principal da aplicação

campo de Boccia, na qual apenas é visível a zona das casas de lançamento, e uma caixa de selecção, do tipo lista. O utilizador apenas necessita de seleccionar o jogador pretendido na caixa de selecção e em seguida escolher a casa de jogada que quer atribuir ao jogador. Esta solução permite que a utilização da interface em causa, se torne mais intuitiva, reduzindo assim o esforço cognitivo do utilizador. Uma outra sugestão, consistiu na reformulação da apresentação das estatísticas, dado que na fase inicial de desenvolvimento apenas foi previsto a geração de estatísticas de uma forma global. A sugestão apresentada foi, que as mesmas pudessem ter um carácter temporal, ou seja, pudessem ser geradas, quer globalmente, quer mensalmente, de forma a poder ser gerida a evolução dos atletas, e para que também possa existir um trabalho específico que incida sobre aspectos menos bons dos mesmos.

3.3 Implementação

Com base nos protótipos de baixa fidelidade foi implementado um protótipo de alta-fidelidade da aplicação direccionado para plataformas do tipo Tablet PC (Figura 7). Praticamente todas as funcionalidades encontram-se implementadas, incluindo a geração de estatísticas de jogo (Figura 8), estando prevista uma avaliação do sistema em contexto real com utilizadores do CPCB. Nesta primeira avaliação serão propostas aos utilizadores tarefas que deverão ser executadas e monitorizadas. As métricas utilizadas nesta avaliação e as formas de registo a aplicar estão neste momento a ser estudadas.

Na implementação do sistema foi utilizado o Microsoft Visual Studio 2003 com a linguagem de programação C#.

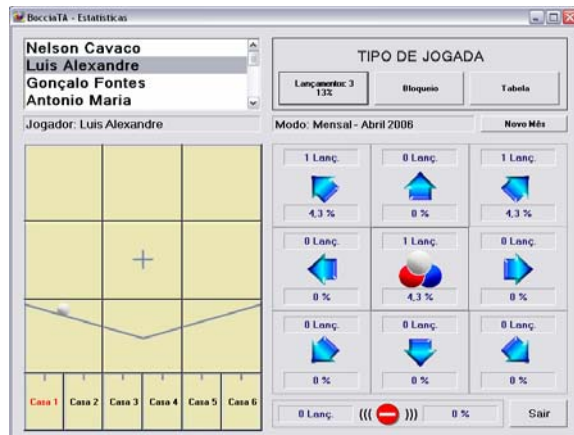


Figura 8 – Ecrã de visualização de estatísticas

4. CONCLUSÃO E TRABALHO FUTURO

A aplicação sistemática de técnicas de desenho (análise de tarefas e prototipagem) revelaram-se extremamente eficazes no desenvolvimento do sistema. A avaliação constante do sistema, quer através do confronto com princípios e regras de usabilidade, quer com os utilizadores permitiu a identificação das melhores opções de desenho. Está neste momento desenvolvido um protótipo funcional do sistema que irá ser testado junto dos utilizadores, após estes testes está prevista a melhoria e a introdução de novas funcionalidades, que permitam um melhor desempenho e a interligação do sistema com outros softwares.

5. REFERÊNCIAS

- [Bailey96] Bailey, Robert. Human Performance Engineering. Prentice Hall. 1996
- [CP-ISRA05] <http://www.bocciainternacional.com/Rules/rulesptfeb05.pdf>. 2005
- [Dix03] Dix, Alan et al. Human-Computer Interaction. Prentice Hall. 2003
- [Nielsen00] Nielsen, J.. Designing Web Usability: The Practice of Simplicity. New Riders Publishing. 2000.
- [Shneiderman04] Shneiderman, Plaisant. Designing the User Interface: Strategies for Effective Human-Computer Interaction 4/E. Addison-Wesley. 2004.
- [Soccer95] www.soccer-software.com, All Stats Software. 1995

Desenvolvimento de uma Aplicação para a Visualização de Sessões de Hidrocinesioterapia

Elizabeth S. Carvalho
DSI, UM
Campus de Azurém, Guimarães
ecarvalho@dsi.uminho.pt

Adérito F. Marcos
DSI, UM
Campus de Azurém, Guimarães
marcos@dsi.uminho.pt

Sumário

A hidrocinesioterapia ou fisioterapia aquática constitui um conjunto de técnicas terapêuticas fundamentadas no movimento humano dentro da água. De forma a maximizar a eficiência da execução, monitorização e avaliação dessas sessões terapêuticas é necessário recorrer a um fato aquático específico, que o paciente veste, provido de sensores electrónicos que permitem registar e monitorar os movimentos natatórios durante a sessão de tratamento. Nesse ponto surge a necessidade do desenvolvimento de uma aplicação computacional que seja capaz não só de armazenar toda a informação recolhida através dos sensores, mas principalmente implementar uma forma conveniente de monitorização gráfica com base na sua visualização clara, intuitiva e concisa, seja em tempo real ou a posteriori, que funcione como auxiliar de diagnóstico e acompanhamento para o terapeuta. Este artigo apresenta uma aplicação em desenvolvimento para a visualização mencionada.

Palavras-chave

Visualização científica, visualização da informação, hidrocinesioterapia.

1. INTRODUÇÃO

A hidrocinesioterapia é a fisioterapia na água [Fiorelli02], ou a prática de exercícios terapêuticos em piscinas, associada ou não a manipulações, hidromassagem e massoterapia, configurada em programas de tratamento específicos para cada paciente. Muitas são as condições patológicas ou sequelas beneficiadas pelo tratamento hidroterapêutico ou hidrocinesioterapêutico: reumáticas, ortopédicas e traumatológicas, neurológicas, respiratórias, cardíacas, endócrinas e psíquicas.

De forma a tirar o máximo partido dessas sessões de hidrocinesioterapia torna-se necessário recorrer a um fato especial com o objectivo de auxiliar a respectiva monitorização e execução. Para este efeito, ele além de ser um fato aquático desenhado para permitir o controle do movimento corporal do paciente, inclui sensores para monitorar alguns parâmetros básicos fisiológicos e motores, como a frequência respiratória e cardíaca, além de movimentos das articulações dos ombros e ancas e flexão da coluna vertebral.

Portanto, é fundamental e necessário o desenvolvimento e implementação de uma aplicação computacional que dê o devido suporte a informação obtida durante a utilização deste fato especial numa sessão de hidrocinesioterapia.

2. OBJECTIVOS

Entre os muitos objectivos existentes numa aplicação desta natureza, podem-se destacar os seguintes como sendo principais:

- Validar a aplicação de metáforas visuais típicas da área de visualização de informação, nesta área do conhecimento.
- Monitorar parâmetros de frequência respiratória, batimento cardíaco, flexão de coluna, movimento de ancas e ombros de pacientes.
- Avaliar esses parâmetros, comparando inclusive com resultados anteriores (avaliação histórica).
- Promover e facilitar o diagnóstico com base nas avaliações.
- Visualizar resultados e avaliações, facilitando a compreensão de resultados.
- Melhorar o acompanhamento do estado clínico do paciente em resposta as sessões de hidrocinesioterapia.
- Minorar possíveis riscos.
- Promover uma interface agradável e fácil para o Terapeuta.
- Processar em tempo real a informação oriunda de vários fatos em simultâneo.
- Gerir o repositório de gravações das sessões.

3. MODELO CONCEPTUAL

O modelo foi concebido e está a ser desenvolvido segundo uma filosofia orientada a objectos. A figura 1 mostra o diagrama de caso de estudo que ilustra o cenário preten-

dido como solução em alto nível para o problema existente. Como pode ser visto, os seguintes actores estão sempre presentes numa sessão de hidrocinestoterapia: Terapeuta, Paciente, Sensores e Aplicação. Dentro do âmbito deste trabalho, será dado apenas enfoque ao actor Aplicação.

Segue-se uma breve descrição das funcionalidades, modelo visual e arquitectura pretendidas.



Figura 1: Diagrama do caso de estudo do problema

3.1. Funcionalidades

A partir do fato será possível obter registos de valores relativos a frequência cardíaca e respiratória, a flexão da coluna e rotação das articulações das ancas e ombros do paciente durante a execução da sessão. Toda essa informação será utilizada pela maior parte das funcionalidades a serem disponibilizadas.

Por outro lado, existem dois aspectos de grande importância para os Terapeutas em termos de análise das sessões de hidrocinestoterapia: a avaliação e monitorização em tempo real da sessão e o acompanhamento da evolução da performance do paciente ao longo do tempo, a posteriori.

A análise, em ambos os casos, apresenta características muito semelhantes, divergindo apenas em termos temporais – a sessão é dinâmica e se baseia apenas nos valores momentaneamente recebidos, enquanto que em termos de histórico, os valores são oriundos de sessões ocorridas em datas diferentes. Numa fase inicial, serão utilizados os valores máximos, mínimos, médios, além das variações e distribuição ao longo do tempo dos valores obtidos pelos sensores numa frequência de 75 Hz, como base para a execução das análises.

A monitorização se caracteriza pela avaliação cíclica dos principais sinais fisiológicos (frequências cardíaca e respiratória) e de movimento das articulações, que indicam se o paciente pode estar a entrar em alguma situação de risco durante a sessão. Caso isso ocorra, deve ser acionado algum sinal de alarme sonoro e visual.

Outro pormenor é a importância da fiabilidade dos registos que são tratados pela aplicação. Apesar de ser feita a filtragem do sinal na entrada (electronicamente), eliminando possíveis ruídos, está também prevista a execução de uma filtragem suplementar dos valores recebidos com base no desvio auferido em relação a média obtida a cada

5 minutos (período de tempo configurável pelo utilizador). Desta forma é garantida uma eliminação satisfatória de possíveis erros. Um outro pormenor é o volume de registos recebidos. É necessário executar alguma redução através do agrupamento dos dados em intervalos de tempo variáveis (granularidade indicada pelo utilizador).

Para além das funcionalidades que permitem a análise da sessão actual e da evolução histórica, a aplicação ainda deverá proporcionar algumas funcionalidades típicas de gestão de base de dados, já que será necessária a existência de uma para gerir a informação pertinente aos registos de pacientes e terapeutas, bem como das respectivas sessões gravadas.

Por último, é importante que o Terapeuta possa facilmente configurar a aplicação de forma que os métodos de análise possam ser devidamente ajustados conforme os objectivos da avaliação, e os aspectos de interacção e visualização estejam de acordo com as preferências e/ou necessidades. Para isso a aplicação deverá permitir uma fácil parametrização e ser passível de fornecer um certo grau de adaptabilidade em termos visuais e funcionais, consoante o caso.

Portanto, as principais funcionalidades a serem incorporadas pela aplicação são:

- Gestão de registos de pacientes e gravações de sessões.
- Análise dinâmica em tempo real de sessões.
- Monitorização do estado físico do paciente durante a sessão.
- Análise da evolução histórica.
- Gravação das sessões.
- Visualização gráfica de resultados e análises.

A figura 2 ilustra as fases que caracterizam o tratamento da informação recebida pela aplicação.



Figura 2: Fases do tratamento da informação

3.1 Visualização

A questão da visualização é um factor relevante para o sucesso desta aplicação. O volume de dados enviado pelos sensores durante uma sessão de hidrocinestoterapia (duração média de meia hora) é impossível de ser efectivamente apreendido sem a utilização de técnicas de visualização expressivas e compreensivas. Embora muitas sejam as técnicas de visualização existentes, cada uma

permite analisar os dados com metáforas visuais diferentes [Plaisant04]. Porém, ao aplicar-se uma determinada técnica de visualização [Taylor00], diversos factores devem ser considerados. A classificação dos dados e das tarefas, a característica cognitiva do potencial utilizador ou o tipo de ênfase visual desejada ao resultado final (conteúdo *versus* quantidade), são alguns exemplos desses factores.

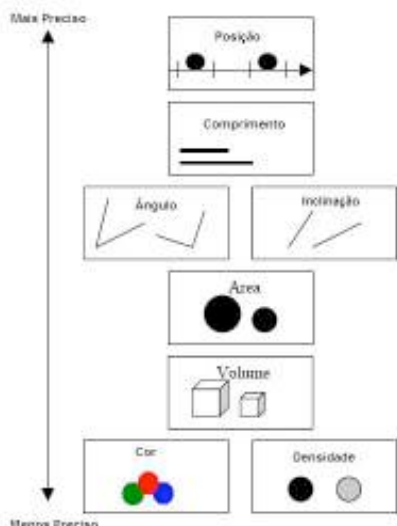


Figura 3: Codificação gráfica segundo Cleveland e McGill's [Cailleteau99]

Um detalhe de importância é a relação entre a precisão dos dados e o volume. A figura 3 mostra a relação proposta por Cleveland e McGill's para a codificação gráfica e a precisão da informação obtida na visualização. Por exemplo no caso da monitorização, é importante que o terapeuta tenha uma clara noção das variações fisiológicas ocorridas (frequência cardíaca e respiratória) no

paciente, porém só é necessário uma maior precisão na visualização (o valor auferido em si) quando ocorre uma situação anormal. Portanto, em termos gerais as visualizações desta aplicação devem ter um ponto de equilíbrio em termos do grau de detalhe, já que o volume de informação é grande, além de permitir uma granularidade de detalhe variável, conforme a solicitação e necessidade do terapeuta.

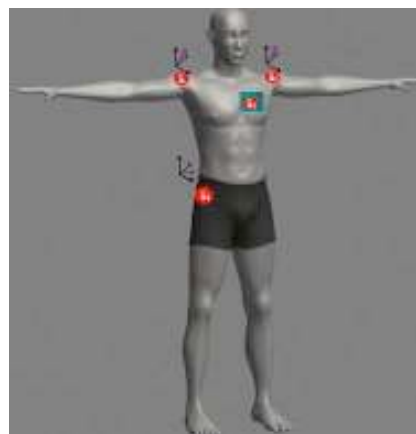


Figura 4: Monitorização da sessão

Por outro lado, há situações em que é interessante visualizar graficamente mais de uma informação em simultâneo. Por exemplo, a análise das rotações das articulações deve ser vista não só em termos de valores máximos, mínimos e médios registados, mas também permitir uma visão vectorial do movimento.

Uma característica de interesse na visualização [Rhyne03] desta aplicação é a conjugação entre técnicas de visualização de informação e científica. Existem funcionalidades cujos resultados pertencem ao domínio da primeira e outras que pertencem ao segundo.

Tarefa	Dados	
Monitorização da sessão	<ul style="list-style-type: none"> - Volumétrica - Símbolos - Pseudo cor - Vector - Gráfico histograma, linha e scatter plot com projecção geométrica - Som - Treemap e scatter plot (2D) 	<ul style="list-style-type: none"> - Modelo 3D do corpo humano (figura 4) - Localização dos sensores no fato - Frequência cardíaca e respiratória - Rotação das articulações - Frequência cardíaca e respiratória, rotação de articulações (projecções para histogramas e gráfico de linhas). - Frequência cardíaca e respiratória (alarme) - Distribuição de valores máximos, mínimos e médios de frequências e rotações - Variações entre valores máximos e mínimos de frequências
Análise da sessão	<ul style="list-style-type: none"> - Gráfico histograma e linha (2D e 3D) - Volumétrica e pseudo cor - Campo de vector 2D 	<ul style="list-style-type: none"> - Modelo 3D do corpo com as articulações coloridas segundo as variações máximas e mínimas de rotação - Distribuição das rotações das articulações
Análise do histórico	<ul style="list-style-type: none"> - Gráfico de área empilhado 3D - Gráfico circular 3D - Gráfico de área 3D + Todas as técnicas propostas para a análise de sessão 	<ul style="list-style-type: none"> - Variações entre valores máximos e mínimos de frequências - Distribuição de rotações - Tendência futura de valores de frequências e rotações

Tabela 1: Mapeamento proposto de tarefas, técnicas e dados

Por exemplo, a geração de gráficos de linha, barras ou *treemap* [Chintalapani04] são técnicas tipicamente pertencentes a visualização da informação. Entretanto, em termos de monitorização das tensões nas articulações do paciente, a utilização de técnicas como a volumétrica e vectores são bons exemplos de técnicas pertencentes ao âmbito da visualização científica.

Com base nas considerações referidas acima e nas necessidades apontadas por uma aplicação desta natureza, a tabela 1 ilustra o mapeamento proposto entre as tarefas, as técnicas visuais e os dados.

3.2 Software e Hardware

Para a implementação da aplicação [VTK06] a nível gráfico será utilizado o VTK 5.0 (*Visualization Toolkit*) da Kitware Inc. (a figura 5 ilustra a arquitectura proposta). Ela é uma ferramenta de código aberto gratuita para a computação gráfica tridimensional, o processamento de imagem e a visualização, consistindo numa biblioteca de classes em C++ e várias camadas de interpretadores incluindo Tcl/Tk, Java e Python. O VTK suporta uma grande variedade de algoritmos de visualização incluindo escalar, vector, tensor, textura, métodos volumétricos, além de métodos avançados de técnicas de modelação. O desenho e implementação desta biblioteca são orientados a objectos.

A nível de interface com o utilizador será [KWWidgets06] utilizado o KWWidgets. Ela é também uma ferramenta de código aberto gratuita para o desenvolvimento de interfaces gráficas, passível de instalação em muitas plataformas, incluindo Windows, Mac e vários tipos de Unix/Linux. Ela foi implementada em C++ e inclui todos os *widgets* padrão normalmente disponíveis – botões, caixas de diálogo, combo-boxes, listas, etc., além de avançadas *widgets* compostas – histogramas, editor de cor, barras de ferramentas, escala progressiva, etc.



Figura 5: Arquitectura da aplicação

Para a implementação da base de dados será utilizado o Access da Microsoft, como linguagem de desenvolvimento o C++, e sistema operativo Windows.

A nível de hardware será adoptado um PC com as seguintes características mínimas de referência: 400 GBytes de disco rígido, processador Intel de 3.4 GHz, placa gráfica GeForce de 256 MBytes.

4. CONCLUSÕES

Em função do elevado volume de informação recebida em tempo real através dos sensores, é fundamental que a aplicação atenda às exigências de processamento com funcionalidades que sejam capazes de gerar visualizações expressivas e eficientes dos resultados obtidos. Aliás, em função desse grande volume, faz-se necessária a combinação de técnicas que permitam uma boa apreensão dos dados bem como atendam de forma conveniente as necessidades cognitivas dos terapeutas e do domínio da área da saúde.

Por último, em função de ser uma aplicação orientada a área da saúde, é de suma importância a fiabilidade e recidivão de resultados e análises tanto a nível visual como funcional, pois só assim estarão garantidas as condições de sucesso de uma aplicação desta natureza e introduzida uma real mais valia.

5. REFERÊNCIAS

- [Cailleteau99] Laurent Cailleteau, Interfaces for Visualizing Multi-Valued Attributes: Design and Implementation Using Starfield Displays, *HCIL Technical Report n° 99-20*, Setembro-1999.
- [Chintalapani04] Gouthami Chintalapani et al. Extending the Utility of Treemaps with Flexible Hierarchy. *Proceedings 8th International Conference on Information Visualization*, 2004, pgs. 335-344.
- [Fiorelli02] Alexandre Fiorelli e Eduardo Arca., Hidrocioterapia: princípios e técnicas terapêuticas, *Coleções Plural*, EDUSC, 2002, Brasil.
- [KWWidgets06] KWWidgets, Kitware Inc., <http://www.kwwidgets.org>.
- [Plaisant04] Catherine Plaisant. The Challenge of Information Visualization Evaluation. *Proceedings of Advanced Visual Interfaces 2004*, Maio-2004. pgs. 109-116.
- [Rhyne03] Theresa Rhyne. Information and Scientific Visualization: Separate but Equal or Happy Together at Last. *IEEE Visualization 2003*, Outubro-2003, pgs. 611-614.
- [Taylor00] Russel M. Taylor II, "New Visualization Techniques", *ACM SIGGRAPH*, vol. 34, n° 1, Fevereiro/2000.
- [VTK06] VTK. Visualization Toolkit, Kitware Inc., <http://www.vtk.org>.

Estudos com Observadores para Avaliação de Qualidade: Influência do Contexto e do Tipo de Dados Avaliados

Samuel Silva
IEETA
Universidade de Aveiro
sss@ieeta.pt

Carlos Ferreira
DEGEI – Universidade de Aveiro
CIO – Universidade de Lisboa
cferreira@egi.ua.pt

Joaquim Madeira Beatriz Sousa Santos
DETI / IEETA
Universidade de Aveiro
{jmadeira,bss}@det.ua.pt

Resumo

Os estudos com observadores são uma possível forma para a validação e teste quer de interfaces de utilizador, quer de sistemas de visualização. Contudo, apesar das inúmeras directrizes propostas na literatura, devido a diversos factores inerentes a essas experiências, o seu desenho é um processo complexo. Com base nos resultados obtidos em dois estudos com observadores, para avaliação da qualidade de modelos poligonais, ilustra-se o efeito de dois factores particulares (contexto e tarefas não-isomórficas).

Palavras-Chave

Estudos com observadores, Malhas Poligonais, Qualidade

1. INTRODUÇÃO

Os estudos com observadores são, hoje em dia, uma prática frequente para a validação e teste de interfaces de utilizador e sistemas de visualização [Tea05]. Um relatório recente [Johnson 06] reafirma os processos de avaliação como fundamentais para determinar o sucesso e caracterizar a aplicabilidade de novas técnicas.

A literatura [Cochran 83, Dea04] tem proposto diversos conjuntos de regras que permitem o desenvolvimento de protocolos de avaliação para os mais variados fins, de forma a que estes possibilitem a recolha de informação relevante e, tanto quanto possível, independente de factores como a ordem das tarefas, aprendizagem e fadiga dos observadores. No entanto, existem factores que podem influenciar uma avaliação (introduzindo desvios nos resultados obtidos) e que muitas vezes não são considerados, tais como [North 06]:

- A escolha das tarefas e a forma como as questões são formuladas;
- O contexto proporcionado a uma tarefa pelas outras tarefas envolvidas na avaliação;
- No desempenho de várias tarefas, à partida equivalentes (isomórficas), as diferenças entre os dados avaliados em cada uma delas;
- Forçar os utilizadores a articular uma resposta pode tornar difícil o tratamento dos dados resultantes, mas

disponibilizar um conjunto de respostas possíveis pode levar o utilizador a decidir usando um processo de eliminação.

A variabilidade introduzida por estes factores torna o desenvolvimento de estudos com utilizadores um processo mais complexo e pode traduzir-se, por ex., na necessidade de aumentar o número de participantes de forma a conseguir maior significância estatística dos dados.

O presente artigo apresenta alguns resultados obtidos para a avaliação da qualidade percebida de modelos poligonais, através de duas experiências de avaliação com observadores, que permitem confirmar a importância de dois dos factores acima referidos.

Começa-se com uma breve descrição do protocolo utilizado para efectuar a avaliação da qualidade dos modelos poligonais de forma a permitir uma compreensão de todo o processo. Depois, alguns dos resultados obtidos são apresentados e discutidos. Finalmente, são apresentadas algumas conclusões.

2. ESTUDOS COM OBSERVADORES

Foi desenhada uma experiência com observadores de forma a avaliar a qualidade percebida de modelos poligonais. Esta experiência consistia em duas fases distintas:

- Fase 1 — Preferências — Os observadores eram con-



Figure 1. Modelos usados na primeira experiência (Conjunto A). O modelo pulmonar mais à direita foi também usado na segunda experiência.

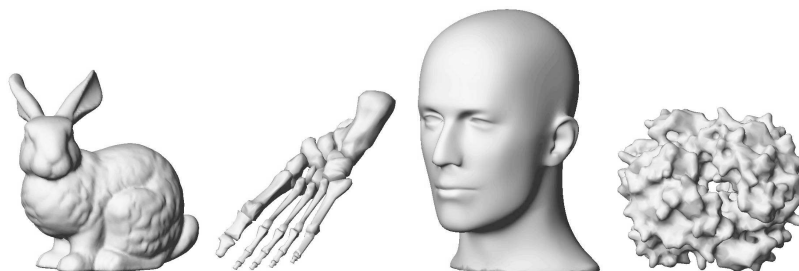


Figure 2. Modelos usados na segunda experiência (Conjunto B). Do conjunto fazia também parte o modelo pulmonar apresentado à direita na figura 1.

frontados com o modelo original e várias versões do mesmo, resultantes de processos de simplificação usando diferentes algoritmos, para um mesmo nível de simplificação (20% e 50% do número original de faces) sendo-lhes pedido que ordenassem (1^o, 2^o ou 3^o) cada um dos modelos simplificados de acordo com a sua qualidade comparativamente ao original.

- Fase 2 — Classificações — Os observadores eram confrontados com um modelo original e uma sua versão simplificada sendo-lhes pedido que classificassem a qualidade da mesma comparativamente ao original usando uma escala de Likert de 5 níveis.

Durante estas duas fases foi também recolhida, para cada observador, informação referente ao número de interações com os modelos (rotações, translações, etc.) e ao tempo despendido para tomar as decisões.

A experiência foi executada para dois conjuntos distintos de modelos poligonais: um conjunto de modelos pulmonares (constituído pelos modelos apresentados na figura 1 — conjunto A) tendo participado 32 observadores; e, posteriormente, um conjunto de modelos de diferente natureza (constituído pelos modelos apresentados na figura 2 e também pelo modelo apresentado mais à direita da figura 1 — conjunto B) tendo participado 65 observadores.

Uma descrição mais detalhada do protocolo usado e dos resultados globais obtidos pode ser encontrada em [Silva 05, Silva 06].

3. RESULTADOS

O facto de os conjuntos de modelos usados em cada uma das experiências conterem um modelo comum permitiu re-

alizar a avaliação da sua qualidade enquadrando-o em diferentes contextos: no primeiro caso somente com modelos pulmonares (conjunto A), no segundo com modelos de diferente natureza (conjunto B). Os resultados apresentados foram obtidos usando o *software* STATISTICA [Sta06].

Começando pelos dados recolhidos durante a primeira fase das duas experiências, foi feita uma comparação dos tempos de decisão e do número de interações registados e referentes ao modelo pulmonar (e suas versões simplificadas). Para os tempos de decisão não se notou nenhuma diferença significativa. No entanto, para o número de interações, como se pode verificar pela análise dos diagramas de extremos e quartis [Hea83] apresentados na figura 3, e para ambos os níveis de simplificação (20% e 50%), o número de interações registadas é aproximadamente o dobro quando o modelo pulmonar se encontra inserido no conjunto B, do que na experiência em que apenas foram usados modelos pulmonares (conjunto A). A significância estatística deste resultado foi confirmada utilizando um teste não paramétrico de Mann-Whitney [Gibbons 97] ($p = 0.0004 \ll 0.05$ e $p = 0.002 \ll 0.05$ para os níveis de simplificação de 20% e 50% respectivamente) que rejeitou a hipótese nula (igualdade das medianas). De notar que não se registou nenhuma diferença significativa, entre as duas experiências, nas preferências atribuídas pelos utilizadores.

De seguida foram analisados os dados obtidos para o modelo pulmonar na segunda fase das experiências. O tempo de decisão foi menor para as situações em que o modelo pulmonar estava inserido no conjunto B (figura 4), tendo sido rejeitada a hipótese de igualdade de medianas através

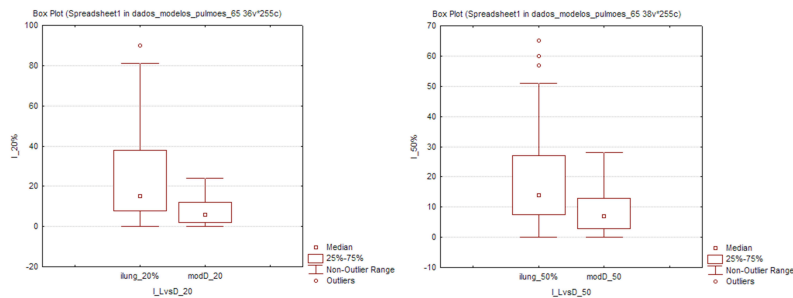


Figure 3. Número de interações registadas para o modelo comum, na fase das preferências, e para os níveis de simplificação de 20% (à esquerda) e 50% (à direita): em ambos os casos o diagrama da direita corresponde à experiência usando apenas modelos pulmonares (conjunto A).

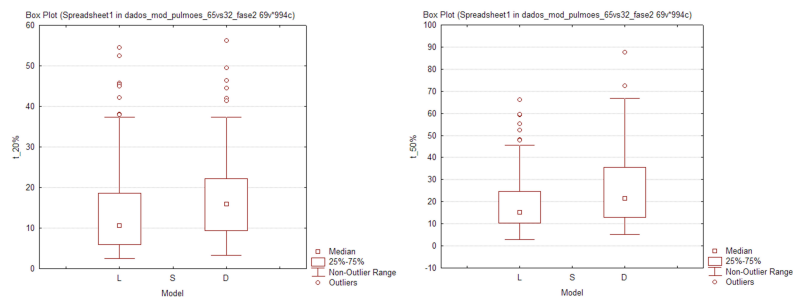


Figure 4. Tempos de decisão registados para o modelo comum, na fase das classificações, e para os níveis de simplificação de 20% (à esquerda) e 50% (à direita): em ambos os casos o diagrama da direita corresponde à experiência usando apenas modelos pulmonares (conjunto A).

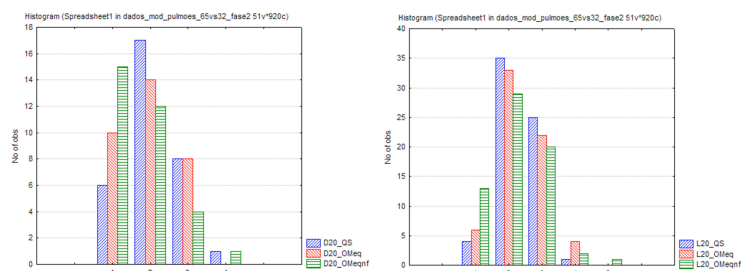


Figure 5. Histogramas das classificações obtidas pelas três versões do modelo pulmonar comum às duas experiências, para um nível de simplificação de 20%: o histograma da esquerda corresponde à experiência usando apenas modelos pulmonares (conjunto A).

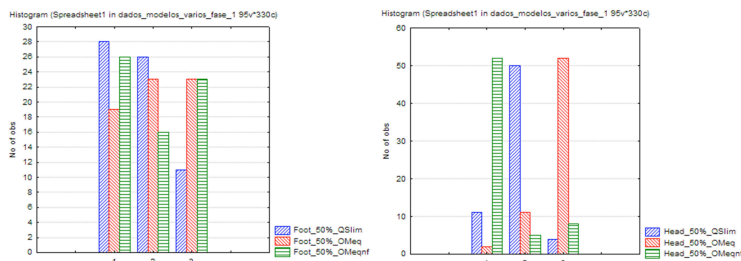


Figure 6. Histogramas das preferências atribuídas aos três algoritmos de simplificação avaliados para o modelo do pé (à esquerda) e para o modelo da cabeça (à direita), para um nível de simplificação de 50%.

de um teste de Mann-Whitney: $p = 0.000169 \ll 0.05$ e $p = 0.000281 \ll 0.05$ para os níveis de simplificação de 20% e 50%. Para o número de interações verificou-se uma igualdade para o nível de simplificação de 20%; para o nível de simplificação de 50% os observadores interagiram menos com o modelo pulmonar quando este se encontrava inserido no conjunto B.

Finalmente, comparando os histogramas das classificações obtidas pelas três versões simplificadas do modelo pulmonar comum em ambas as experiências (figura 5), verifica-se que os modelos simplificados obtiveram globalmente melhores resultados aquando da segunda experiência: o número de classificações mais baixas é proporcionalmente menor.

A análise dos dados referentes à segunda experiência [Silva 06] revelou um comportamento diferente do modelo da cabeça em relação aos outros modelos. Enquanto que para os outros modelos o tempo de decisão aumenta com o nível de simplificação, com a cabeça passa-se o contrário, sendo os observadores mais rápidos para o modelo menos simplificado. Para além disso, os histogramas das preferências do modelo da cabeça para os dois níveis de simplificação têm uma particularidade não observada nos dos outros modelos e que pode ser percebida, por exemplo, pela análise dos histogramas apresentados na figura 6: note-se que o histograma da cabeça (à direita) apresenta uma mais clara distinção entre os métodos de simplificação do que o referente ao modelo do pé (similar ao dos outros modelos).

4. CONCLUSÕES

Os resultados apresentados demonstram a importância de factores que nem sempre são considerados no desenho e análise dos resultados obtidos numa experiência com observadores. A análise dos dados resultantes da avaliação da qualidade do modelo poligonal do pulmão, obtidos nas duas experiências, permitiu verificar que, de facto, o diferente enquadramento deste nos dois conjuntos de modelos influenciou os resultados obtidos, nomeadamente ao nível dos tempos de decisão, do número de interações e das classificações obtidas. O facto de os observadores interagirem mais com o modelo pulmonar quando este estava inserido no conjunto A parece indicar uma análise mais cuidada por parte dos observadores (mais exigente) o que explicaria os valores mais baixos obtidos nas classificações. A presença de modelos de um único género parece ajudar a que os observadores foquem a sua atenção em detalhes dos mesmos (analisando-os com mais cuidado) que de outra forma passariam despercebidos.

Os resultados mais interessantes surgiram para o modelo da cabeça (segunda experiência) e demonstram um comportamento diferente deste modelo comparativamente aos outros. O tempo de decisão mais pequeno e a diminuição do mesmo para os modelos menos simplificados (contrariando os resultados globais obtidos de um tempo de decisão menor para modelos mais simplificados) podem ser motivados pela elevada capacidade do cérebro humano de processar informação de faces [Gregory 98] e pelo facto

de a análise de faces ser uma tarefa usual no dia a dia, o que treina o observador sobre as características principais a analisar. Para além disso, o modelo da cabeça permitiu discriminar, de forma mais clara, entre os métodos de simplificação, o que levanta a questão sobre quais as suas características que motivam este comportamento, e também se o modelo exibe a mesma capacidade de discriminação para outros métodos de processamento.

Apesar de obtidos num contexto particular e carecendo de uma análise mais cuidada, os resultados apresentados demonstram que é importante considerar os detalhes relacionados com o contexto em que as tarefas são desempenhadas, já que estes podem influenciar os resultados obtidos. Demonstram também que é necessário algum cuidado no desenho de tarefas que se pretendem isomórficas (diferentes instâncias do mesmo tipo de tarefa) sob pena de, apesar de formalmente equivalentes, apresentarem resultados influenciados pela natureza dos dados.

5. AGRADECIMENTO

O primeiro autor agradece à Unidade de Investigação 127/94 IEETA, da Universidade de Aveiro, a bolsa que vem financiando o seu trabalho.

BIBLIOGRAFIA

- [Cochran 83] W. G. Cochran. *Planning and Analysis of Observational Studies*. John Willey, 1983.
- [Dea04] A. Dix et al. *Human Computer Interaction*. Prentice Hall, 3ª edição, 2004.
- [Gibbons 97] J. D. Gibbons. *Nonparametric Methods for Quantitative Analysis*. American Sci. Press, 3ª edição, 1997.
- [Gregory 98] V. Gregory. *Eye and Brain, the Psychology of Seeing*. Oxford Univ. Press, 5ª edição, 1998.
- [Hea83] D. Hoaglin et al. *Understanding Robust and Exploratory Data Analysis*. John Wiley & Sons, 1983.
- [Johnson 06] C. Johnson, et al. *NIH / NSF Vis. Research Challenges Rep.* IEEE Press, 2006.
- [North 06] C. North. Toward measuring visualization insight. *IEEE Comp. Graph. & App.*, 26(3):pp. 6–9, 2006.
- [Silva 05] S. Silva, et al. Comparing three methods for simplifying mesh models of the lungs: an observer test to assess perceived quality. Em *Proc. SPIE 2005, vol 5749*, páginas 484–495, 2005.
- [Silva 06] S. Silva, et al. Perceived quality of simplified polygonal meshes: Evaluation using observer studies. Em *Proc. Ibero-American Symp. in Comp. Graph. SIACG06*, 2006.
- [Sta06] Statistica 6.0. <http://www.statsoft.com>, 2006.
- [Tea05] M. Tory et al. Evaluating visualizations: Do expert reviews work? *IEEE Comp. Graph. & App.*, 25(5):pp. 8–11, 2005.

Evaluating an Activity-centered Ubiquitous Computing Framework for User Assistance in Public Spaces

Helder Pinto Rui José

University of Minho, Information Systems Department
Campus de Azurém, 4800-058 Guimarães, Portugal
{helder, rui}@dsi.uminho.pt

Abstract

This paper describes a ubiquitous computing framework for supporting activities that are strongly associated to a specific physical environment and that are performed by occasional visitors. Our framework implements an activity-centered approach to ubiquitous computing, by defining a conceptual model inspired by Activity Theory and implementing a software infrastructure derived from this conceptual model. The framework has been evaluated from the user view-point by experiments run at different public spaces. We are proposing to perform an additional experiment with the framework, providing user assistance to Interacção 2006 participants and organizers.

Keywords

Ubiquitous computing frameworks, activity-centered computing.

1. INTRODUCTION

Public spaces accommodate many kinds of users and activities. Some people are recurrent users of such spaces (e.g., local workers) while others go there occasionally for short-term activities (e.g., conference participants). The activities that can be performed in public spaces vary between activities which relevance or interest is not associated to the physical environment (e.g., managing e-mail or editing a report) and activities that can only be physically achieved in a specific place (e.g., visiting an exhibition at a museum or visiting a relative at the hospital). The latter are called *localized activities* given that they are strongly related to a specific physical location. Particularly interesting are localized activities performed by *occasional visitors*, i.e., people that are not used to live or work in that place and that occasionally pass by. Before these visitors come to some public space, they have little or no idea about the physical setting nor about the resource infrastructure that such an environment may provide to support localized activities. These users need anchors that help them to easily orient in the physical environment and to identify and use the resources (humans or artifacts) available for achieving the activity. Public spaces in general are designed to provide such assistance to occasional visitors, for example, by the means of wall signs, panes, brochures, etc. However, these means lack some important characteristics such as personalization, full availability, completeness, or interactivity. Moreover, these means are seldom integrated with each other and are often designed with a functional perspective rather than oriented towards the possible activities each particular visitor may achieve.

We believe that ubiquitous computing can greatly enhance the experience of public space visitors, by offering effective and transparent means for achieving their localized activities, while providing a personalized assistance, treating users as individuals with different motivations, context, and preferences. This work is also based on the conviction that an activity-centered approach to ubiquitous computing is a promising path for bringing computing closer to people and to transparently support activities that take place in the physical world [Banavar 00, Christensen 02, Sousa 02, Norman 05]. Taking an activity-centered approach for system design becomes especially important in situations in which people have little or no prior knowledge about the physical environment or about the local means available for the activity they are going to perform. This approach faces several challenges:

- how to model human activity and user interaction independently of the different mental models of activity that people may have;
- how to represent this model in a machine-understandable manner, simple enough to require minimum specific know-how and effort from public space administrators managing the local ubiquitous computing infrastructure;
- how to deal with the heterogeneity of interaction devices a ubiquitous computing environment may supply while not compromising simplicity of user interaction;

- how to deal with the possibly varied interaction devices the same person may use within the course of an activity and to make that person feel that all interactions, whatever device is used, are integrated and all part of the same activity;
- and personalization issues like availability of and access to personal data or privacy issues.

In the following section, we present Activity Theory, the theoretical framework upon which we ground our conceptual model for representing activities and user interaction. Section 3 describes this conceptual model and presents ActivitySpot, the activity-centered software framework we propose for supporting localized activities. Section 4 describes the evaluation objectives of our demonstration in the conference. Finally, section 5 concludes the paper.

2. MODELLING ACTIVITY AND USER INTERACTION

We believe that the best approach to overcome the activity and user interaction modelling challenges is to ground our research on previous work on human activity analysis. The importance of a theoretical framework of human activity is that it provides ubiquitous computing researchers with an agreed set of terms to describe activity and with concepts that drive them in the construction of systems that intend to support activity. Among several frameworks produced mainly by the fields of psychology and philosophy, we chose Activity Theory [Leontiev 81] as the background for this work, based on its maturity acquired by several decades of research and its set of simple and solid concepts. Among these concepts, we are particularly interested in the different levels of analysis of an activity: *activities*, at the uppermost level, are distinguished on the basis of their motive and the object toward which they are oriented; *actions* are distinguished on the basis of their goals; and, finally, *operations*, on the basis of the conditions under which they are carried out. For example, an activity motivated by food is composed of several goal-oriented actions (e.g., collecting ingredients, preparing a recipe, etc.) and operations which vary in function of conditions (e.g., going to the kitchen-garden, picking vegetables, taking ingredients from the fridge, etc.).

An activity may be carried out in a variety of ways by employing different actions and operations, which may respectively be part of different activities and actions. Individual characteristics and changing local and personal context are the factors driving the structure of a localized activity. For example, a public space like a museum may support different activities, which in turn may employ different actions and operations, all depending on several factors, like the visitor role (e.g., regular museum visitors, authors, external security inspectors, etc.), age, preferences, available resources, or context. In a ubiquitous computing environment supporting these hypothetical localized activities, each ubiquitous computing device can be seen as a tool that may be used for the execution of one or more

operations. Figure 1 depicts our Activity-Theory-inspired abstract model of activity applied to the museum example.

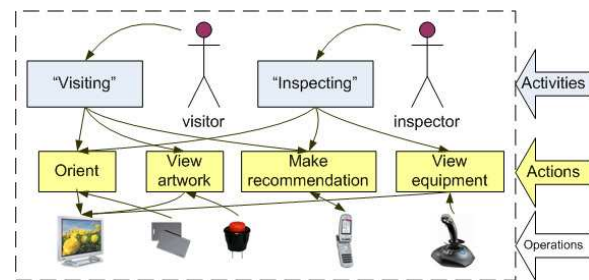


Figure 1. Example of an activity-based model for a ubiquitous computing environment

For the sake of clarity, the model omits the details of operations. In a ubiquitous computing system, an operation can be a user interaction, a sensor read, a web-service request, a database query, etc. We just represent the user-facing devices, which are the most visible part of operations. The model exemplifies how flexible an activity structure can be: a plasma screen can be used both by visitors and inspectors to achieve different actions; an “orient” or a “make recommendation” action can be executed in different activities, with different goals in mind (a recommendation made by a visitor has a different goal from a recommendation made by an inspector).

Given that user interaction with a ubiquitous computing system is done through multiple, heterogeneous means, we reduced user interaction analysis to basic human-computer interaction concepts: stimulus and response. We assume that, for a given stimulus through a given device, a response is produced, synchronously or not, through the same device or through other device or set of devices. We also assume that people interact with ubiquitous computing systems mainly through simple devices. We consider a simple interaction device in a ubiquitous computing environment as being the equivalent of a mouse, a keyboard, or a screen in a desktop computer. We are talking about elementary, easy-to-use interaction means that cannot be used only by themselves to carry out an activity. The execution of an activity is thus distributed by the interactions made with each of those devices. Every user interaction, whatever the underlying medium, is framed within the user activity and is integrated with other previous and further interactions, becoming more meaningful and contributing to compose the whole activity.

3. THE ACTIVITYSPOT FRAMEWORK

The ActivitySpot framework provides a set of conceptual and software tools for designers and developers applying an activity-based approach for supporting occasional visitors to ubiquitous computing environments. The concepts basing the framework are derived from Activity Theory, namely those associated to the activity structure analysis. Therefore, the principles of the ActivitySpot framework originated from the concepts of activity, action, and op-

eration, as well as the activity structure hierarchy and flexibility. We also consider that activities or actions depend on local and personal context, either as an execution condition or as a variable influencing the response of an operation. Finally, the framework includes the basic concepts of stimulus and response to model user interaction. The conceptual model is implemented in the architecture described in Sect. 3.2.

The main strength of this conceptual model, besides its activity-centered character, is its simplicity. Such a simple model leads to simple architectural abstractions and, therefore, to an easier task for public space administrators using ActivitySpot for supporting localized activities.

3.1. Environment Specification

In order to be independent of physical space and activities and thus support any localized activity scenario, the ActivitySpot framework is based on a generic specification format for activities, actions, and interaction devices available in an environment. Each environment supported by ActivitySpot has a specification of: a) which actions can be executed – name, supported stimulus and response types, a reference to the component implementing the action controller, and execution conditions; b) which activities are available – name, execution conditions, and references to the actions composing it; and c) which local devices can be used – stimulus or response type, physical location, and references to other devices which have some physical or logical association.

Activity specification is currently done by means of an XML document. Future developments of ActivitySpot will include a graphical user interface providing high-level abstractions easing the generation of the activity specification.

3.2. The ActivitySpot architecture

The ActivitySpot architecture (see Fig. 2) implements the activity-centered conceptual model described earlier. Following the generic character of the environment specification, the ActivitySpot architecture provides abstractions powerful enough to be instantiated in several concrete scenarios and simple enough to facilitate the adoption and usage by public space managers without requiring deep computer science know-how. The main architecture component is the Activity Manager. It manages activity execution by coordinating stimuli, execution conditions verification, and response generation. Another crucial component is the EQUIP [Greenhalgh 02] data-space, which is used as a communication middleware between interaction devices and Activity Manager.

The Activity Manager, following the environment specification, knows the characteristics of each supported activity and respective actions. The Activity Manager is also responsible for launching action controllers – the components implementing action logic¹ – and listening for in-

¹The logic implemented in the action controllers corresponds to the set of operations that constitute the respective actions. The operational logic may be shared between action controllers.

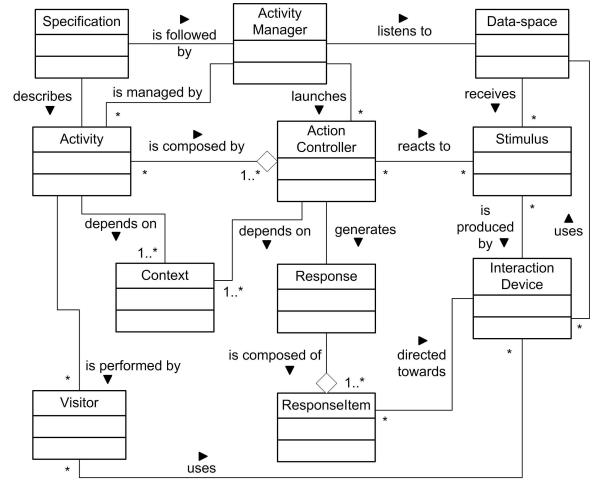


Figure 2. The ActivitySpot architecture

teraction events in the data-space. Data collected from visitors is kept in a relational database. We assume that visitors, previous to the system usage or during the activity unrolling, provide the association to their personal domain. We currently support *ad-hoc* associations mechanisms, e.g., providing the mobile phone number through an initialization message or associating RFID tags or a Bluetooth address to a visitor at a registration desk. The aimed generalizable association mechanisms, which would be used by visitors wherever they go, are going to be investigated in future work.

Whenever a visitor generates a stimulus through an interaction device, a corresponding stimulus description is sent to the data-space. The Activity Manager senses this stimulus, identifies its author (e.g., a mobile phone number, a MAC address, an RFID code, etc.), and triggers all the action controllers that fill the execution conditions and that support the respective stimulus type. Each of these action controllers processes the stimulus sent by the Activity Manager and, in the case the stimulus was effectively targeted to the respective action², a corresponding response is produced – a response may be composed of one or several response items directed towards specific interaction devices. The Activity Manager sends the response items to the data-space, which propagates them to the interaction device presenting that response type. When several devices may consume the same response type, the Activity Manager is able to address the response item to the device that is physically closer to the user. This is achieved when the Activity Manager is able to derive the user location from the stimulus. For example, if an RFID tag carried by the user is intentionally brought near a reader, the user is expecting to see the response in a nearby display, not in a display elsewhere.

The stimulus reaction behavior is similar to what happens for an event generated by a context sensor. Actions that are

²In order to avoid ambiguity, we assume that a given stimulus (e.g., a specific RFID tag, an SMS message with a specific format, etc.) is always targeted to a single action

sensible to context changes may thus generate a response to an interaction device or, if a response is not suitable, execute some logic without producing any response.

4. EVALUATION

Although ActivitySpot has been demonstrated in different public space scenarios [Pinto 06], we are seeking to evaluate it in a scenario that better meets our objectives. A conference scenario such as *Interacção 2006* has characteristics that make it a particularly interesting place for demonstrating ActivitySpot: it provides at least two different activities – participating in the conference and collaborating in the conference organization; both activities are performed by people that generally are not used to the physical environment nor to the activity performed in that particular place; both activities may share common actions and operations (e.g., viewing the list of participants), demonstrating a central concept of our work; both activities may be performed by a control group without ubiquitous computing assistance and compared to ActivitySpot users. Furthermore, conference participants and organizers would certainly be interested and active evaluators of our framework. The demonstration will be held during the whole conference and intends to provide assistance both to conference attendees and organizers in their respective activities.

5. CONCLUSION

This paper presents ActivitySpot, a ubiquitous computing framework for supporting localized activities performed by occasional visitors to public spaces. This work is based on an activity-centered approach for system design, which becomes especially important in situations in which people have little or no prior knowledge about the physical environment or about the activity they are going to perform. The main contributions of this work are the Activity Theory-inspired conceptual model and a software infrastructure, derived from this model, providing a generic tool set for ubiquitous computing environments supporting localized activities. We are proposing to evaluate ActivitySpot in the *Interacção 2006* conference, by providing user assistance to *Interacção 2006* participants and organizers, which is a particularly interesting and rich scenario meeting our evaluation requirements.

6. ACKNOWLEDGEMENTS

This work has been financially supported by “Fundação para a Ciência e a Tecnologia” under grant SFRH/BD/13299/2003 and is partially supported by FEDER through POS_Conhecimento.

References

- [Banavar 00] Guruduth Banavar, James Beck, Eugene Gluzberg, Jonathan Munson, Jeremy Sussman, and Deborra Zukowski. Challenges: An application model for pervasive computing. In *Sixth ACM/IEEE International Conference on Mobile Networking and Computing*, pages 266–274, Boston, USA, August 2000. ACM Press.
- [Christensen 02] H.B. Christensen and J. Bardram. Supporting human activities – exploring activity-centered computing. In *Ubiquitous Computing 2002 (UbiComp 2002)*, volume 2498, pages 107–116, Göteborg, Sweden, September 2002. Springer-Verlag.
- [Greenhalgh 02] C. Greenhalgh. EQUIP: a software platform for distributed interactive systems. Technical Report Equator-02-002, Department of Computer Science, University of Nottingham, 2002.
- [Leontiev 81] A. N. Leontiev. The problem of activity in psychology. In J. Wertsch, editor, *The Concept of Activity in Soviet Psychology*, pages 37–71. Sharpe, Armonk, USA, 1981.
- [Norman 05] D. Norman. Human-centered design considered harmful. *Interactions*, 12(4):14–19, July-August 2005.
- [Pinto 06] H. Pinto and R. José. Activity-centered ubiquitous computing support to localized activities. In *Conference on Mobile and Ubiquitous Systems*, pages 119–128, Guimarães, Portugal, June 2006.
- [Sousa 02] J. P. Sousa and D. Garlan. Aura: an architectural framework for user mobility in ubiquitous computing environments. In *3rd Working IEEE/IFIP Conference on Software Architecture*, pages 29–43, Montreal, Canada, August 2002. Kluwer Academic Publishers.

Interior Design Tool for Common Users

Alfredo Ferreira Roberto Medeiros Rui Barreto Manuel J. Fonseca Joaquim A. Jorge

Intelligent Multimodal Interfaces Group
Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon

<http://immi.inesc-id.pt>

Abstract

Until recently, computer simulation of interior decoration was reserved to architects and other highly skilled professionals, because existing tools require technical knowledge on computer aided design. During the last years, several easy-to-use tools for home planning and interior design have been released. These tools intend to allow users, without specific skills, to create their virtual house, decorate it and navigate in the resulting scenery. In this paper we present three of these applications and describe task analysis using them. As a result of this experiment, we identify user requirements for this kind of tools, along with positive and negative aspects, providing good insights and directions for future work.

Keywords

Computer-Aided Design, Sketch-Based Interfaces, User Interfaces, Task Analysis, Evaluation

1 Introduction

There are several software tools for interior design. However, most of them are directed for architects, interior designers or for users with a good knowledge in CAD applications. These type of tools are very hard to use by common users, because they require a long learning time and because they force users to acquire specific skills on interior design and/or in CAD applications.

To understand the motivations, skills and limitations of potential users, and also to identify the main features that this kind of applications should have, we carried out task analysis. It was not our intent to perform an exhaustive evaluation of existing applications, but only to identify user requirements in order to establish guidelines for future development of this kind of tools. Since there are several applications, in the market and in research laboratories, to do interior decoration, we decide to perform task analysis using only a small set. We selected the *Punch! Professional Home Design Suite* [PUN], the *IKEA Office Planner* [IKE] and the *DecoSketch* [Brito 05] prototype. During task analysis we asked users to perform a set of tasks in the three applications, while we videotaped everything and took note of their comments and suggestions. The main conclusion from task analysis was that existing systems

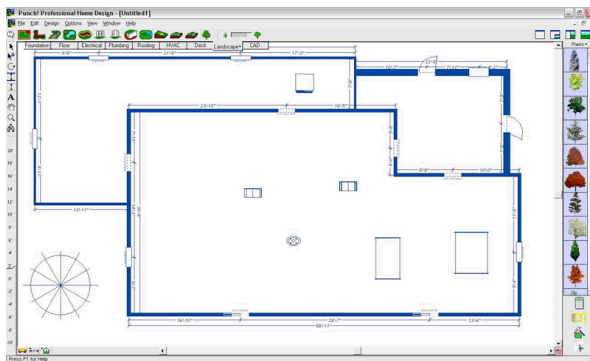
were not designed for common users. They were created for professionals with, at least, some knowledge of architecture or interior decoration in conjunction with some experience of computer assisted drawing.

To overcome these limitations, we plan to develop an interior design tool based on calligraphic interfaces, recognition techniques and 3D retrieval mechanisms. Our solution will offer common users an easy, simple and expedite way to create the floor plan and decorate the divisions of their houses, by inserting decorative elements stored in a database of furniture.

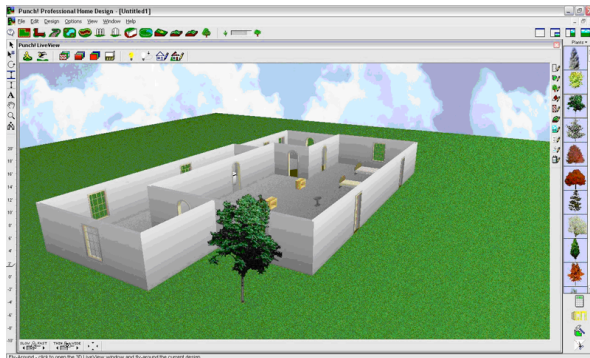
The rest of this paper is organized as follows: next section provides an overview of three tools for interior design. In section 3 we describe the task analysis and present the main results achieved. Finally, we discuss our conclusions and present directions for further development.

2 Interior Design Tools

Currently, several tools for interior design are available as commercial applications or research prototypes. Most of these tools were created for expert users, namely architects or other highly qualified professionals. However, some recent applications have been developed to allow users without specific skills to design their own home plan. From



2D editor



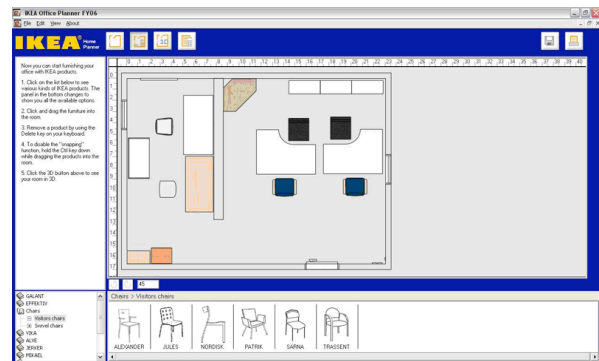
3D viewer

Figure 1. Punch! Professional Home Design

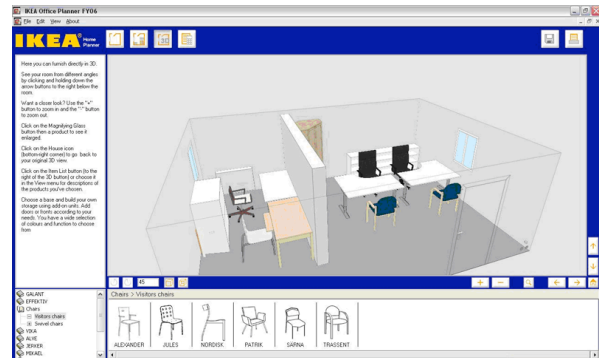
these tools we selected a small set to experiment with final users. With such experiments we intended to identify user requirements for this kind of tool and identify the positive and negative points of selected tools, and not to perform user evaluation.

Interior design applications can be classified in three distinct categories: commercial tools for professional or semi-professional use; commercial applications provided by furniture manufacturers or suppliers to their customers; and laboratory prototypes that generally feature novel approaches and methodologies. For our experiment, we selected three different applications, one from each of these categories. The *Punch! Professional Home Design Suite (Puch-HD)* is a commercial application. The *IKEA Office Planner (Ikea-OP)* is a free tool that customers of a furniture manufacturer can download and use. Finally, the *DecoSketch* is a research prototype of a calligraphic tool for home decoration. In the following paragraphs we will shortly describe each one of these applications.

The *Punch-HD* (see Figure 1) is a very complete and professional program, with a complex and not so obvious interface for beginners. It integrates a wide range of functionalities besides interior design, such as realistic 3D rendering or furniture creation. This application can satisfy both home design professionals and common users. We choose this tool, because it has a relative success in the market, so we can get some good ideas from it, and also because we wanted to see how fast a common user would learn how to use it.



2D editor



3D viewer

Figure 2. Ikea Office Planner

The main objective of *Ikea-OP* (see Figure 2) is to allow easy and quick creation of a house plan, and to put furniture inside to see the resulting scenery. Additionally, the user can know the overall cost of his configuration (users can only include furniture from the manufacturer catalogue). We chose this system because we want to identify its limitations and check its easy of use, since this application was not designed for professionals.

Finally, *DecoSketch* (see Figure 3) is an academic prototype that uses mechanisms we want to include in our solution, such as calligraphic interfaces and sketch recognition methods. We selected *DecoSketch* because we want to validate these ideas and also because we want to see how well users deal with the pen-based interaction in such system.

This task analysis with users, using the three applications allowed us to identify a path for future work in sketch-based interior design research. The experiment with final users, the collected results and its contribution for our work will be described in the following sections.

3 Task Analysis

The main goal of the task analysis was to collect a set of user and functional requirements in order to establish guidelines for the development of novel applications for interior decoration. To that end, we asked seven potential users to perform a set of tasks in the three selected applications. None of these users had previous experience with architectural design tools. We videotaped users in action, as depicted in Figure 4, and annotated comments and suggestions. Additionally, users answered two questionnaires,

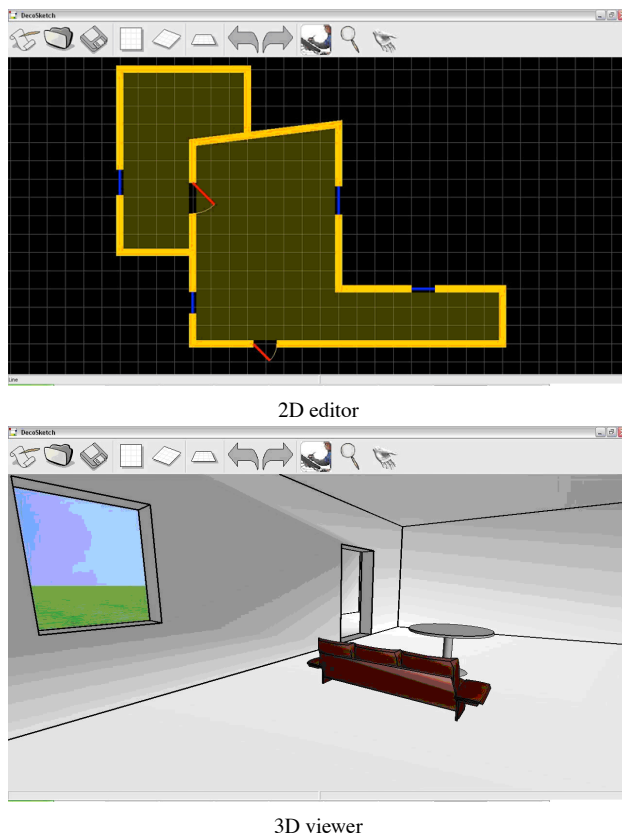


Figure 3. DecoSketch

one at the beginning, to collect information about their profile, and another at the end to measure their satisfaction about the systems.

Each task analysis session was divided in three parts. First, we gave a brief introduction about the experience (objectives, sequences of tasks, documents to fill, etc.) and asked users to fill out the first questionnaire about general data, background, motivation, etc. Second, we presented and explained each application to users, followed by the realization of a set of pre-defined tasks on each system. By defining the same set of tasks for the three applications, we wanted to know what users like and dislike on each system, what mistakes they make and also try to figure out how easily users perform them. In the first task, users must create from scratch a floor plan of a "L"-shaped room with a door and a window. Next, users must navigate in the newly created room, taking a look at the ceiling, the floor and the six walls. Then, users have to place furniture at specific positions inside the room. Finally, users need to add a second window to that room and move furniture to a different position, with a different orientation. After finishing these tasks, we asked users to fill out another questionnaire about the three applications. The first section of the questionnaire collected information about positive and negative aspects of each application individually, while the last section allowed the comparison between all applications according to specific aspects, such as, easiness to draw a plant, easiness to add furniture, etc.



Figure 4. User testing session.

In the next section we present a brief analysis of the answers to our questionnaires, notes taken during the experiment and informal comments produced by users.

3.1 Analysis of Results

The set of tasks performed during task analysis was defined to analyze the three main features of interior decoration tools: floor plan specification; selecting and placing of furniture; and changing existing models.

In general, users were satisfied with the functionalities related to floor plan design, in all tools. Although, different applications offer distinct alternatives to draw the floor plan, these were equally welcomed by users. They liked the freedom of creation provided by *DecoSketch*, the clear manipulation from *Ikea-OP* and the precision from *Punch-HD*. So, a future application must provide all these features.

However, users were not so satisfied with the applications when they need to change or improve the floor plan. For instance, users were displeased that *Ikea-OP* does not allow them to apply colors or patterns to walls. But, users liked the way this application allows them to change the length and position of walls easily. On the other hand, users generally considered changing the floor plan with *Punch! Professional Home Design* a complex task, mostly because of the numerous options offered by this tool.

During the experiment we noticed that users were always expecting to receive visual feedback of every action, sometimes even while they are performing it. For example, when selecting a color or a texture for the floor, users wish to see immediately the final result while browsing in the palette.

The *Ikea-OP* was pointed by users as the best application, while the *Punch-HD* was commonly criticized by its com-

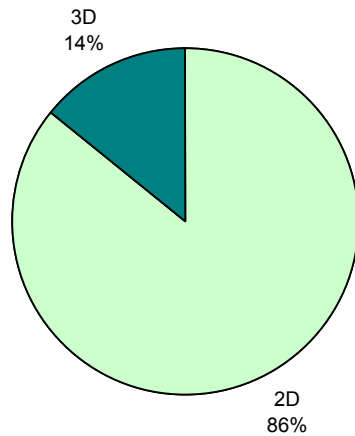


Figure 5. Preferences for floor plan editing.

plexity. Since this last tool is mainly designed for professional users, it is natural that unexperienced users feel uncomfortable using it.

During the tests, *DecoSketch* was often pointed as a second choice tool. Users generally prefer either *Ikea-OP* or *Punch-HD*. However, *DecoSketch* is just a research prototype with very limited functionality, while the other two are commercial, fully functional tools. Nonetheless, according to answers in questionnaires and informal conversations, users were quite pleased with the pen-based interaction and with the navigation mechanisms in both views, 2D and 3D. Thus, we can conclude that the interaction technique proposed by *DecoSketch* should be explored and improved in future work, to produce a fully functional prototype.

It was clear, from the experiments carried out, that most users clearly prefer to edit the plan in 2D rather than in 3D, as depicted in Figure 5. Although, users consider 3D editing useful for some tasks, they are not determinant. Users prefer to edit in two dimensions and then visualize the results in a three dimensional viewer.

After defining the floor plan, users have to place furniture in the created rooms. This task includes the selection of the furniture, its placement and orientation. Commonly, users need to change the displacement of furniture several times during interior design. In this matter, both *IKEA Office Planner* and *Punch! Professional Home Design* satisfies user needs, despite the more complex approach of the latest, mostly caused by the larger number of options. On the other hand, this functionality is not complete in *DecoSketch* prototype. Thus, users cannot fully appreciate the advantages of a sketch-based interface in this task.

When asked if they will use any of these applications again, all users answered positively for *Ikea-OP* and *DecoSketch*. The opinions around *Punch-HD* were not so unanimous. Even though users consider it the more complete and professional tool, less than fifty percent of them consider using it in the future. We believe the main reason for this value is its complexity.

4 Conclusions

From this task analysis, we conclude that *IKEA-OP* has the most important features to replicate in novel interior design tools for common users. It is easy to use and easy to learn. Its simplicity and efficiency pleased users more than the complexity of a powerful tool such as *Punch-HD*. Moreover, users liked the calligraphic interface of *DecoSketch* and point it as the best and more natural interaction approach for quick interior design, due to its resemblance to the pen-and-paper metaphor.

The work described in this paper, along with other work we developed recently, namely the comparison with similar applications and informal conversations with architects, designers and common users interested in this kind of tools, allow us to understand and identify users needs and requirements in this area. We believe that the information collected during task analysis provides good insights and directions for the development of a new tool for interior design.

Acknowledgements

This work was funded in part by the Portuguese Foundation for Science and Technology, project DecorAR, reference POSC/EIA/59938/2004. Alfredo Ferreira was supported by the Portuguese Foundation for Science and Technology, grant reference SFRH/BD/17705/2004.

References

- [Brito 05] Tiago Brito, Manuel J. Fonseca, and Joaquim A. Jorge. Decosketch - towards calligraphic approaches to interior design. In *Proceedings of the International Conference on education and research in Computer Aided Architectural Design in Europe (eCAADe'05)*, September 2005.
- [IKE] Ikea office planner. IKEA: <http://www.ikea.com/>.
- [PUN] Punch! professional home design suite. Punch! Software: <http://www.punchsoftware.com>.

Multi-Modal Painter: Pintar Usando as Mãos e Fala

Mariana Cerejo José Santos Alfredo Ferreira Manuel J. Fonseca Joaquim A. Jorge

Grupo de Interfaces Multi-Modais Inteligentes
 Departamento de Engenharia Informática
 INESC-ID/IST/ Universidade Técnica de Lisboa

<http://immi.inesc-id.pt>

Sumário

Nos últimos anos várias equipas de investigação têm vindo a desenvolver um novo paradigma de interacção entre humanos e computadores, procurando reduzir ou eliminar totalmente a necessidade de manuseamento de dispositivos físicos para controlar a máquina. O trabalho aqui apresentado foca a integração de reconhecimento de gestos e fala para controlo de aplicações. Para validarmos esta integração desenvolvemos uma aplicação multi-modal que consiste numa ferramenta de desenho 2D. Com esta aplicação o utilizador pode desenhar e pintar utilizando apenas gestos e comandos de fala.

Palavra-chave

Interfaces Multi-Modais, Reconhecimento de Gestos, Reconhecimento de Fala

1. INTRODUÇÃO

Para além da evolução tecnológica, a necessidade de dar uma resposta às exigências dos actuais utilizadores, serviu de impulso para um novo conceito de interacção pessoa-máquina. O antigo conceito de WYSIWYG (“What You See Is What You Get”), que colocava o utilizador como agente externo e manipulador, está prestes a deixar de ser a única verdade absoluta neste assunto. Assim, o utilizador deixa de ser a entidade que manipula os dispositivos de introdução de informação, tornando-se ele próprio o “veículo da informação”.

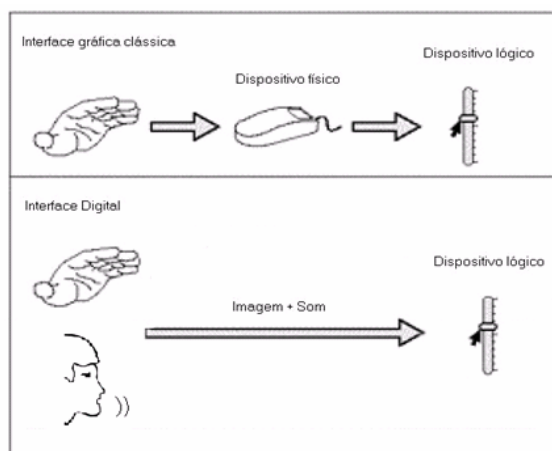


Figura 1 – Mudança de paradigma de Interface

Seguindo esta linha de desenvolvimento, apresentamos um paradigma baseado na utilização de gestos (com as mãos) e comandos de fala para controlar o sistema.

Para realizar os gestos, o utilizador coloca dedais de cores diferentes nos dedos. Utilizando uma câmara captam-se os gestos do utilizador, enquanto que com um microfone, é feito exactamente o mesmo, mas para a fala. Ambos os sinais de entrada são reconhecidos e posteriormente processados e interpretados, de acordo com o contexto corrente.

2. ARQUITECTURA

A Figura 2 representa uma arquitectura geral do sistema. Como se pode constatar, existe uma estruturação em camadas que definem cada módulo. Assim, existe um módulo que trata o reconhecimento gestual, um outro que trata o reconhecimento de fala e o principal que consiste na integração multi-modal dos dois. Além destes, existe ainda a própria aplicação, que faz a fusão multi-modal das duas técnicas de interacção.

Esta arquitectura permite-nos ter um sistema distribuído, podendo ter cada componente em computadores separados, ou seja, se desejarmos podemos ter uma máquina responsável pelas câmaras (componente gestual), outra pelos microfones (componente de fala) e outra responsável pela aplicação.

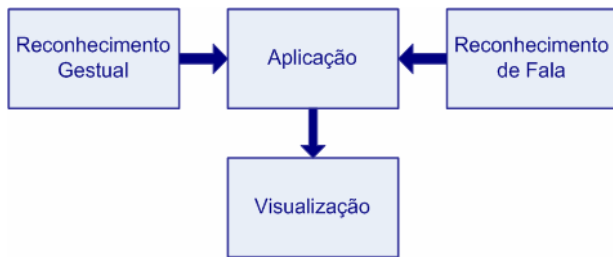


Figura 2 – Arquitectura geral do *Multi-Modal Painter*

Acrescentamos ainda um componente de calibração ao módulo de reconhecimento gestual, para minimizar os efeitos da variabilidade da luminosidade do ambiente na taxa de reconhecimento.

3. RECONHECIMENTO GESTUAL

Uma das mais valias do nosso sistema está no facto de suportar qualquer tipo de fundo, dispensando assim a necessidade de um ambiente controlado para obter boas taxas de reconhecimento. No entanto, é necessário proceder a uma fase de calibração das cores das marcas, para que estas possam ser reconhecidas.

3.1 Calibração

Para o bom funcionamento do sistema é necessária uma calibração prévia sempre que mudamos de ambiente. Para haver um seguimento de marcas robusto foi necessário ter em conta as variações de luminosidade nas diferentes áreas da imagem capturada pela câmara. Assim, a interface utilizador apresenta cinco quadrados (quatro nos cantos da imagem e um no centro) em que o utilizador terá de colocar cada marca, captando toda a informação necessária para se realizar a calibração (ver Figura 3).

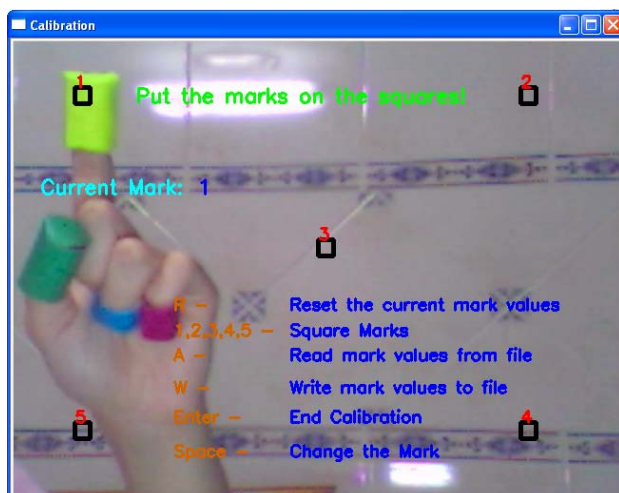


Figura 3 – Calibração inicial das marcas

Esta informação fica guardada num ficheiro, que será posteriormente utilizada pela aplicação, sendo apenas necessário executar a calibração uma única vez, desde que não haja alterações de luminosidade no ambiente, ou movimentos bruscos na câmara.

3.2 Detecção de gestos

A identificação dos gestos realizados pelo utilizador é composta por três passos: primeiro fazemos detecção das marcas que se encontram nos dedos do utilizador; segundo, construímos polígonos ligando os centróides de cada marca (sempre pela mesma ordem); finalmente, identificamos o polígono resultante de modo a reconhecer a pose realizada.

A detecção das marcas durante a interação é baseada em técnicas de visão por computador. Através dos valores obtidos na calibração, aplicamos filtros passa-banda para a tonalidade e passa-alto para a saturação, obtendo-se desta forma a posição das marcas. Em conjunto com a aplicação dos filtros, utilizamos também algoritmos para eliminação de ruído. Identificada a localização das marcas, calculamos os seus centróides, e com base nestes o polígono correspondente.

Para classificar os polígonos resultantes, utilizamos uma biblioteca de reconhecimento de formas geométricas, chamada CALI [Fonseca02]. Esta biblioteca usa um conjunto de características baseadas em áreas e perímetros dos maiores e menores triângulos/rectângulos, criados a partir do polígono original, conseguindo-se uma boa precisão na detecção dos gestos (ver Figura 4).



Figura 4 – Reconhecimento de gestos com o CALI

De modo a identificar os valores de cada característica para cada tipo de pose, realizamos testes com oito utilizadores. Começamos por pedir a cada utilizador que colocasse os dedais (marcas) nos dedos e que efectuasse todos os gestos que se encontram na Figura 5, por ordem. Depois deste passo, voltamos a pedir a cada utilizador que voltasse a repetir as poses, mas agora com o objectivo de medir a taxa de reconhecimento. A taxa de reconhecimento situou-se nos 93%. Estudos posteriores revelaram que a utilização de uma árvore de decisão, para situações em que uma das marcas fica oculta, em combinação com a biblioteca CALI apresenta melhores resultados.

No estudo experimental, procuramos ainda verificar e validar a semântica associada a cada um dos gestos.

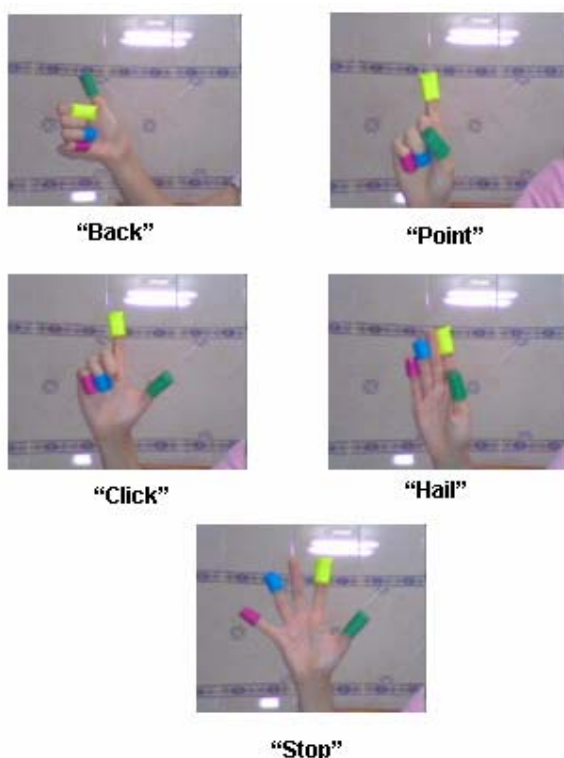


Figura 5 – Gestos Reconhecidos

3.3 Semântica gestual

Para cada gesto estudamos qual seria o melhor significado que lhe poderíamos atribuir, no contexto da nossa aplicação de desenho. Decidimos então atribuir a cada gesto um único evento, conforme descrito na Tabela 1. Para simplificar a aplicação e utilizar o menor número de gestos (para ser mais simples para o utilizador), cada gesto pode ter uma semântica diferente consoante o seu contexto, como podemos ver pelo exemplo do Click. Neste caso, o gesto Click servirá não só para simular o botão esquerdo do rato (para escolha de opções ou de determinados comandos) mas também para efectuar a própria pintura na aplicação.

Tabela 1 – Emparelhamento Gesto-Evento

<i>Gesto</i>	<i>Evento</i>
Back	Botão direito do rato
Point	Controlo do cursor
Click	Botão esquerdo do rato
Hail	Apagar
Stop	<i>Blur</i>

A escolha dos gestos e respectivas características de reconhecimento foram objecto de estudo conforme se pode analisar em [Ferreira06].

4. RECONHECIMENTO DE FALA

A outra técnica de interacção abordada consiste no reconhecimento de fala que, possibilita ao utilizador proferir comandos específicos.

Assim, a contribuição essencial do reconhecimento de fala consiste em permitir ao utilizador dar comandos, mais rapidamente do que usando gestos ou outro modo de interacção mais tradicional. Por exemplo, para efectuar a acção cortar/colar seria necessário efectuar a opção de “seleccionar”, “cortar” mover para o sítio desejado e ainda seleccionar novamente a opção “colar” no sistema de menus. Com a fala esta interacção torna-se muito mais eficaz pois não se fica preso ao sistema de menus. Outro exemplo, um utilizador encontra-se a efectuar um desenho com a cor vermelha e pretende mudar para a cor verde. Tem duas opções:

- Através de gestos, selecciona no menu correspondente, a cor verde;
- Profere a palavra “Green” e a cor é automaticamente mudada.

A tecnologia de reconhecimento de fala adoptada foi o Microsoft Speech SDK, pois a sua taxa de reconhecimento é alta, mesmo sem treino prévio. Além disso, consegue não só reconhecer palavras isoladas mas também frases completas. No nosso trabalho pretendemos apenas identificar comandos isolados.

5. APLICAÇÃO

Para validar o paradigma proposto, criámos uma aplicação multimodal de desenho em 2D que é controlada por gestos e fala (ver Figura 6). De notar que, apesar da nossa escolha para a validação ter recaído sobre esta opção, a nossa abordagem é suficientemente genérica para ser aplicada em alternativas distintas, igualmente válida para o efeito, tais como “browsers” ou modelação.

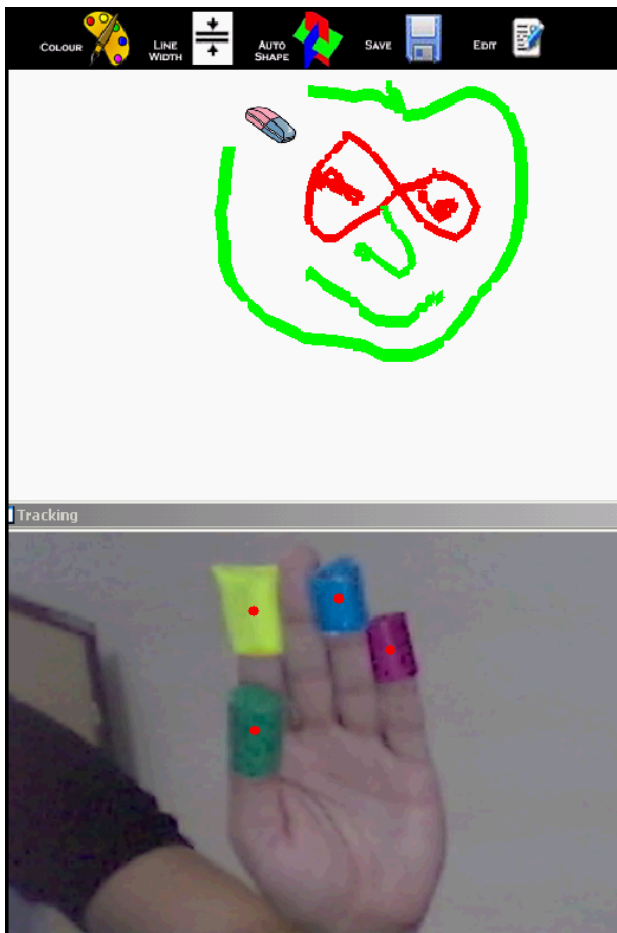


Figura 6 – Multi-Modal Painter

A aplicação desenvolvida permite efectuar um conjunto de acções, quer utilizando apenas gestos, quer usando as duas técnicas. Combinando ambas as técnicas, as limitações de uma são complementadas pela outra e vice-versa. Assim pode-se desenhar, apagar, escolher a cor e a grossura do traço, gravar e abrir o desenho, escolher uma forma predefinida (por exemplo, quadrado ou triângulo), copiar, cortar e colar uma área, preencher, mover e seleccionar um objecto, apagar tudo, fazer *blurring* ou recorrer à ajuda, usando gestos e/ou fala.

6. CONCLUSÕES E TRABALHO FUTURO

O objectivo inicial, que consistia em demonstrar que existem novos paradigmas de interacção que podem substituir os convencionais, torna-se facilmente concreto pois neste caso o utilizador tornou-se veículo de informação não sendo necessário recorrer a periféricos externos para o efeito.

Apesar de ainda não se terem realizado testes formais com utilizadores, os resultados obtidos foram bastante promissores pois consegue-se fazer o controlo da aplicação utilizando apenas gestos e fala sem recorrer aos dispositivos habituais.

Em suma, os resultados obtidos foram bons mas futuramente iremos elaborar uma sessão de testes com utilizadores para obter uma avaliação de usabilidade válida.

Para além da avaliação, pretendemos ainda estender a nossa abordagem para aplicações distintas da aqui apresentada. Nomeadamente, na sua adaptação com vista a obter um sistema de gestão de documentos semelhante ao BumpTop 3D Desktop [Agarawala96].

7. REFERÊNCIAS

[Agarawala96] Anand Agarawala, Ravin Balakrishnan, Keepin' it Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2006)*.

[Ferreira06] Alfredo Ferreira, Ricardo Jota, Mariana Cerejo, José Santos, Manuel J. Fonseca, Joaquim A. Jorge. Recognizing Hand Gestures with CALI, aceite para publicação em *Ibero-American Symposium on Computer Graphics (SIACG 2006)*

[Fonseca02] Manuel J. Fonseca, César Pimentel and Joaquim A. Jorge, CALI: An Online Scribble Recognizer for Calligraphic Interfaces, *Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding*, pages 51-58, Palo Alto, USA, Mar 2002

Plataforma Extensível para Interfaces Multimodais

Tiago Guerreiro¹, Vitor Fernandes², Bruno Araújo³, Joaquim Jorge⁴, João Pereira⁵
IST-UTL / INESC-ID

R. Alves Redol, 9, 1000-029 Lisbon

{tjvg¹, vmnf², brar³}@immi.inesc.pt, jaj@acm.org⁴, jap@inesc.pt⁵

Abstract

Esta comunicação apresenta uma plataforma para a gestão de aplicações multimodais de forma reutilizável e extensível. Para atingir este objectivo focamos a arquitectura nas Capacidades dos dispositivos e nas Intenções dos utilizadores. Abordamos, em particular, ambientes colaborativos onde a integração de várias modalidades e aplicações obriga à existência de um sistema extensível que combine, em tempo real, diversos componentes provenientes de plataformas heterogéneas. Este artigo descreve o enquadramento proposto e demonstra como interligar aplicações e utilizadores em torno de um ambiente imersivo (tiled display wall), incluindo diferentes modalidades de interacção não convencionais e aplicações.

Keywords

Framework, Interfaces Multimodais, Extensibilidade, Reutilização, Capacidades, Intenções, Colaboração.

1. INTRODUÇÃO

Os dispositivos de interacção tradicionais, tais como o rato e o teclado, são ainda esmagadores perante o uso de modos avançados de interacção. No entanto, grandes esforços foram realizados nas últimas décadas com resultados prometedores, para apresentar alternativas na interacção entre pessoas e qualquer tipo de computador. Adicionalmente, existem diversos projectos para interligar modalidades de interacção e construir aplicações multimodais de utilização natural [4]. Apesar da existência de varias modalidades não convencionais, muito poucas aplicações tiram vantagens destas tecnologias devido ao seu custo de implementação [2]. Num ambiente que integre diversas aplicações multimodais, o esforço da equipa de desenvolvimento é frequentemente desperdiçado devido ao foco numa determinada aplicação. A nossa *plataforma* permite ao utilizador gerir tanto as modalidades de interacção como as capacidades das entradas. Com este propósito, focamo-nos na reutilização das capacidades das entradas e na integração de novas modalidades de interacção em aplicações com um esforço reduzido. Por outro lado, permitimos a sua utilização em ambientes colaborativos, onde cada entrada é controlada de acordo com as

preferências da aplicação e com as capacidades disponíveis num determinado momento. Finalmente, para demonstrar a plataforma num contexto colaborativo, apresentamos um cenário imersivo usando uma *display wall* em que várias modalidades de entrada são utilizadas.

2. PLATAFORMA

O principal objectivo da nossa plataforma é gerir modalidades de entrada e aplicações separadamente permitindo assim, para cada componente, a sua reutilização e extensão. A plataforma foca nas capacidades das entradas e nos interesses dos utilizadores de forma a oferecer em cada momento a modalidade de entrada mais adequada a uma determinada tarefa, no contexto de uma aplicação. As capacidades são um conjunto de *tokens* ou dados de entrada que podem ser enviados a qualquer aplicação, isto é, comandos do rato, *tokens* de gesto, coordenadas de posicionamento ou comandos por voz. Por outro lado, um determinado *token* pode ser enviado por diversos módulos de entrada, por exemplo, “cima” poderia ser fornecido por um dispositivo tipo teclado ou por um reconhecedor suportando gestos corporais. A *plataforma* é responsável pelo redireccionamento dos *tokens* dos módulos de entrada para as aplicações de acordo com as suas preferências. As capacidades e as preferências das entradas e aplicações são definidas *a priori* pelo programador/utilizador. Desta forma, diferentes modalidades podem ser integradas e partilhadas entre aplicações. Além disso, uma nova entrada oferecendo um *token* já previamente conhecido por uma aplicação, pode ser facilmente integrada estendendo as suas modalidades sem alterações no lado da aplicação.

2.1 Visão Geral do Sistema

Para atingir os objectivos de extensibilidade utilizámos

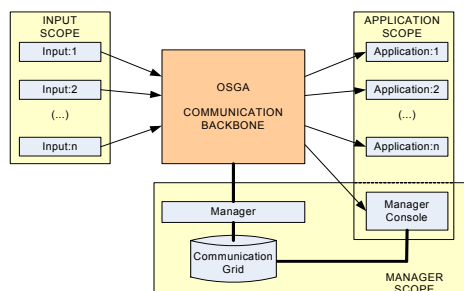


Figure 1 – Arquitectura do Sistema

uma abordagem baseada em mensagens. A arquitectura do nosso sistema, apresentada na Figura 1, está organizada em quatro entidades diferentes: Módulos de entradas, Aplicações, um Gestor e uma infra-estrutura de comunicação denominada OSGA[5]. Os módulos de entrada são as interfaces de comunicação para dispositivos simples tais como o teclado e rato mas também sistemas de reconhecimento de voz ou sistemas baseados em seguimento de câmaras que conseguem entregar tokens multimodais correspondentes a comandos por fala, gestuais ou outro tipo de dados tais como ficheiros. As aplicações estão ligadas à infra-estrutura de comunicação através de uma interface que permite que a interacção tire partido da informação transmitida pelos módulos de entrada. A infra-estrutura de comunicação é responsável pelo redireccionamento de todas as mensagens, permitindo a inter-comunicação entre vários módulos de entradas e múltiplas aplicações. A comunicação e a descoberta de novos módulos de entrada é coordenada por uma quarta entidade que tem a função de gestor. Este módulo de Gestão suporta a configuração e o estabelecimento de comunicações entre módulos de entrada e aplicações. A coordenação consiste num mecanismo de ligação matricial que guarda as capacidades disponibilizadas por cada módulo de entrada activa e as necessidades ou preferências de cada aplicação. Finalmente, o Gestor oferece uma interface que permite ao utilizador seleccionar módulos de entrada alternativos em relação à preferência da própria Aplicação. Esta interface, tal como as aplicações, beneficia de todos os módulos de entrada disponíveis num dado momento por parte da plataforma.

2.2 OSGA: Open Source Group Architecture

Existindo vários protótipos resultantes dos trabalhos de investigação em curso, originou a necessidade de planear trabalhos futuros de integração que permitam juntar pequenos projectos e criar aplicações integradas de larga escala. A OSGA (Open Software Group Architecture)[5] é uma plataforma de integração distribuída baseada em mensagens XML desenvolvida no nosso grupo de investigação para ultrapassar o problema da integração. Esta plataforma pode ser facilmente utilizada em aplicações futuras bem como integrada com protótipos actuais/passados. Foi desenvolvida sobre o módulo XmlBlaster[6] e providencia a existência de vários clientes a receber mensagens de acordo com mecanismos de subscrição e publicação. Basicamente, todas as mensagens enviadas para o sistema são re-direccionadas para todos os clientes que subscreveram a um dado tópico de mensagem. O sistema suporta igualmente, filtrar mensagens de acordo com propriedades adicionais tais como a identificação do remetente, ou indícios sobre o conteúdo da mensagem.

2.3 Protocolo de Comunicação

Para suportar a interacção entre módulos de entrada, aplicações e o gestor, organizamos o nosso protocolo de mensagens XML (Figura 2) em quatro fases distintas. A primeira fase é a inicialização das entradas, que permite

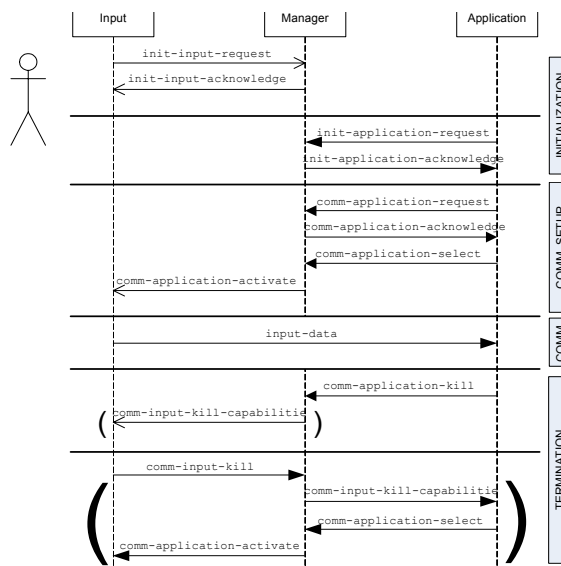


Figure 2 – Diagrama de Sequência (Protocolo)

registar no gestor novos módulos de entrada assim como suas capacidades. É igualmente possível, registar as capacidades desejadas de cada aplicação de forma a identificar os modulos de entrada que suportam tais capacidades. A terceira fase é a configuração de uma comunicação para estabelecer um canal de comunicação entre uma dada entrada e uma aplicação com o objectivo desta receber dados de entrada. Finalmente, o protocolo suporta a suspensão/re-estabelecimento de um canal de comunicação entre uma entrada e uma aplicação e sua finalização que é coordenada pelo Gestor.

Inicialização: A inicialização de um módulo de entrada é executada pela mensagem *init-input-request* usando uma canal provisório de comunicação. Como ilustrado na Figura 3, cada entrada fornece a sua identificação assim como a lista de capacidades suportadas pelo módulo de entrada. A identificação é formada pelo tipo do dispositivo (exemplo: Fala, EMG, Dispositivo de Seguimento...) e a chave que será usada pelo gestor para atribuir um identificador global (exemplo Fala:1) ao modulo de entrada. Esta informação é armazenada no gestor, e a nova identificação é utilizada pelo módulo de entrada de forma a criar um canal que estará disponível para as aplicações interessadas. As aplicações, tal como descrito na Figura 2, requerem também uma inicialização afim de serem identificadas pelo gestor. Por outro lado, a lista de capacidades requisitadas é fornecida e armazenada na matriz de comunicação do Gestor. O Gestor define um identificador global (exemplo Paint:1) e fornece uma lista dos módulos de entrada e das capacidades disponíveis para satisfazer as necessidades da aplicação. A Figura 3 apresenta um exemplo da resposta em *init-application-acknowledge*. Esta informação é usada posteriormente pela aplicação de forma a estabelecer a comunicação com os módulos de entrada necessários para suportar a interacção desejada.

Comunicação e Interacção (Figura 4): As aplicações podem obter, a qualquer momento, a lista das entradas e

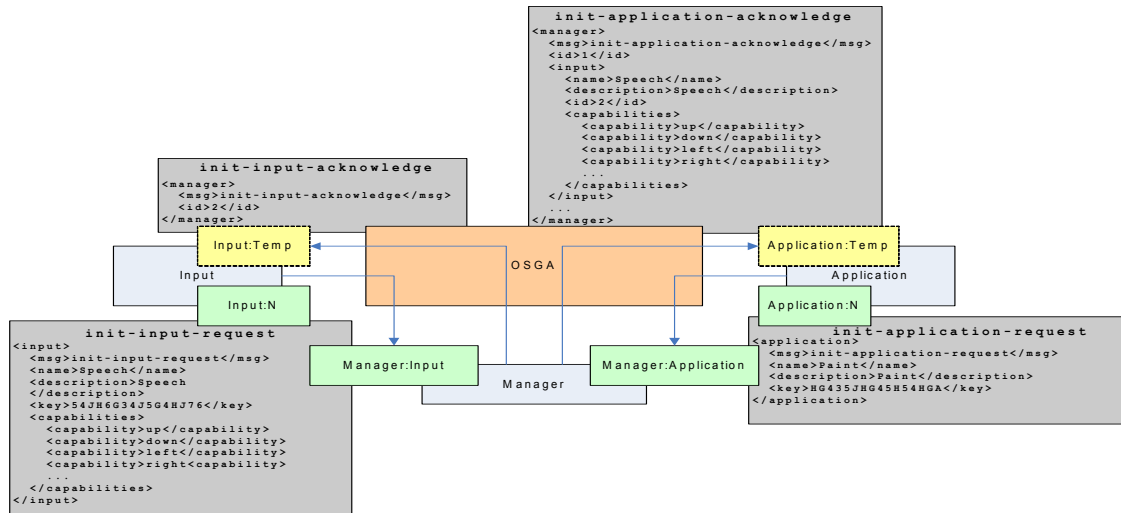


Figure 3 – Diagrama de Inicialização

capacidades (comm-application-request), e são informadas pelo gestor do estado actualizado da plataforma (comm-application-acknowledge). Com essa informação, as aplicações podem seleccionar as capacidades conforme as preferências dos utilizadores como pode ser visto no seguinte exemplo XML.

Application->Manager:Application(Filter: <none>)

```
<application>
  <msg>comm-application-select</msg>
  <name>Paint</name>
  <id>1</id>
  <input>
    <name>Speech</name>
    <id>2</id>
    <comm-type>XML</comm-type>
    <capabilities>
      <capability>up</capability>
      <capability>down</capability>
    </capabilities>
  </input>
  ...
</application>
```

O gestor é responsável por comunicar aos módulos de entrada (com as capacidades requisitadas) que existem aplicações que estão interessadas em receber seus dados correspondente a uma dada capacidade. Desta forma, uma entrada só publica os dados que foram requisitados, isto é, no cenário de duas aplicações A e B em que a aplicação A quer “esquerda”, “direita” e “seleccionar” e a aplicação B quer “acima”, “abaixo” e “seleccionar”, só as entradas “esquerda”, “direita”, “acima”, “abaixo” e “seleccionar” serão publicadas dado que correspondem a união de dados que foi requisitado por cada aplicação. Cada aplicação recebe exclusivamente os dados que requisitou tirando partido da funcionalidade de filtragem oferecida pela OSGA. A comunicação é estabelecida entre os módulos de entradas e as aplicações que subscreveram os tópicos dessas entradas. Utilizando esse

mecanismo de subscrição, consegue-se uma comunicação de n:n entre módulos de entrada e aplicações. Os mecanismos de filtragens permitem uma melhor regulação do tráfego de mensagens graças ao uso de filtros e descrições ricas (mensagens trocadas) que podem ser actualizadas a qualquer momento. Os módulos de entrada publicam as mensagens identificadas pelo tópico cuja a nomenclatura é a seguinte: {tipo da entrada}:{identificação}:data.

Input -> Paint:1:Data (Filter:up,down)

```
<input>
  <msg>input-data</msg>
  <data>up</data>
</input>
```

Suspender e Continuar: A comunicação entre um módulo de entrada e uma aplicação pode ser suspensa por iniciativa da aplicação através do cancelamento da subscrição da entrada. Para restabelecer a comunicação, basta subscrever novamente o tópico desejado. Caso um módulo de entrada termine durante a suspensão de uma comunicação, o Gestor informa a aplicação do sucedido (comm-application-acknowledge). A aplicação pode opcionalmente comunicar ao gestor que suspendeu a comunicação de uma entrada. Essa informação é registada pelo Gestor.

Terminar: As aplicações podem terminar a sua sessão em qualquer altura desde que publiquem essa intenção ao Gestor (comm-application-kill). O Gestor informa os módulos de entradas caso não haja mais aplicações a requisitar um determinado dado. Dessa forma, é possível evitar tráfego de mensagens desnecessário. Por exemplo, se a entrada estiver a publicar “para cima” e “para baixo” mas a única aplicação que estava interessada em “para cima” terminou então a entrada começa publicar apenas “para baixo”. Quando uma entrada termina a comunicação (comm-input-kill), o Gestor informa as aplicações afectadas que têm de modificar a sua dependência de entradas/capacidades. A lista actual das entradas e respectivas capacidades é então enviada para as aplicações efectuarem uma nova selecção (como em comm-application-select).

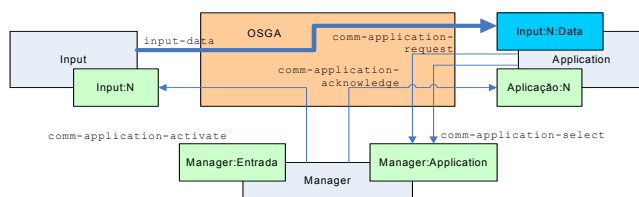


Figure 4 – Diagrama de Comunicação

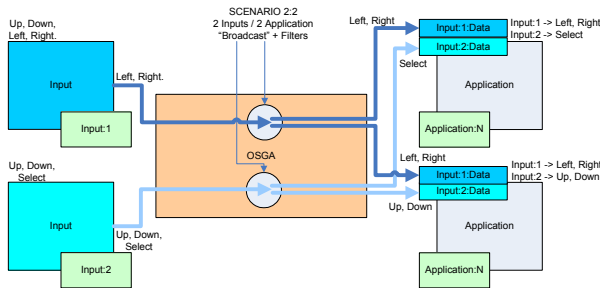


Figure 5 – Cenário colaborativo

Keep-alive: Para detectar situações de falha na comunicação dos módulos de entrada ou das aplicações com o gestor quanto a sua terminação, desenvolveu-se uma sinalização *keep-alive* entre essas entidades e o gestor. Se uma falha for detectada o protocolo Terminar será seguido dependendo da entidade onde foi detectada a falha.

3. CONTEXTO DE UTILIZAÇÃO

A plataforma apresentada neste artigo é a base para todos os protótipos de interação e aplicação futuros no nosso ambiente imersivo, LEMe Wall [1]. Assim, para além de possibilitar uma estrutura colaborativa, possibilitamos também a reutilização de todos os protótipos de interação criados noutros contextos. Além do ecrã de larga escala existente, o ambiente é complementado por um conjunto de sensores e actuadores que permitem aumentar a imersão. Através de vários dispositivos de interação não convencionais (Sensores de pressão, sensores de posicionamento, câmaras para reconhecimento de gestos e microfones para reconhecimento de voz) podemos ter diversos utilizadores a interagir simultaneamente com o ambiente no mesmo ou em diferentes contextos/aplicações. A utilização da nossa plataforma neste ambiente permite que cada módulo de Entrada criada seja reutilizado por qualquer investigador que desenvolva uma aplicação e torna fácil adicionar novas Entradas e Aplicações ao leque de componentes existente.

3.1 Cenários de Aplicação

A Figura 5 apresenta um cenário com duas aplicações e duas Entradas. Com a descrição deste cenário, pretendemos demonstrar as vantagens da estrutura apresentada, contemplando duas entradas diferentes que controlam uma aplicação, comandos diferentes de uma entrada usados por duas aplicações simultâneas e também comandos de uma entrada que afectam ambas as aplicações. Consideremos um cenário colaborativo onde dois utilizadores estejam a desenhar um modelo na LEMe Wall. As Entradas seleccionadas são o TabletPC e os gestos, operados pelos utilizadores. A Aplicação usada é o Gides++[3], que foi desenvolvida no nosso grupo de investigação. Com a nossa abordagem, os utilizadores podem operar dentro da mesma aplicação simultaneamente, o que aumenta a colaboração e o desempenho do processo de desenho (Figura 6). Assim, os utilizadores podem interagir sobre a mesma aplicação mas focando em aspectos ou tarefas diferentes. Vamos



Figure 6 –Desenho Colaborativo na LEMe Wall

agora considerar um cenário com duas aplicações, um visualizador de informação geográfica e um visualizador de imagens satélite de alta-resolução. Cada utilizador controla uma das aplicações através de uma determinada Entrada. Considerando que os utilizadores operam sobre a mesma área e escala dos respectivos mapas, é possível ter comandos de uma Entrada utilizados por ambos e que afecte ambas as aplicações. Assim, neste cenário, considerando as acções de Zoom ou Pan, conseguimos manter o sincronismo entre as duas aplicações. Este exemplo contempla comandos enviados para uma aplicação e para duas aplicações independentes. Os cenários apresentados podem ser realizados simultaneamente criando redes mais complexas de comunicação.

4. CONCLUSÕES

Apresentámos uma estrutura para gerir aplicações multimodais em ambientes distribuídos focando na extensibilidade e reutilização das modalidades de entrada. Atingimos este objectivo separando entradas e aplicações e controlando a sua comunicação através de um protocolo baseado em mensagens. A contribuição principal deste trabalho é a possibilidade para adaptar em tempo real as entradas e aplicações conforme as capacidades disponíveis e a vontade do utilizador num determinado momento. São apresentados cenários colaborativos motivadores que demonstram as possibilidades da plataforma.

5. REFERENCES

- [1] Araújo, B., et al. Leme Wall: Desenvolvendo um sistema de Multi-projecção. Proceeding of 13^o EPCG, Vila Real, Portugal, Out 2005.
- [2] Flippo, F., et al. A framework for rapid development of multimodal interfaces. In *Proc. of the 5th ICMI 03*. ACM Press, New York, NY, 109-116
- [3] Jorge, J.,Silva, N., Cardoso, T. Gides++, Proceedings of 12^o EPCG, Porto, Portugal, Out 2003
- [4] Krahnstoever, N., et al. *A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays*. in 4th IEEE ICMI 2002.
- [5] <http://www.osga.net>, Junho 2006
- [6] <http://www.xmlblaster.org>, Junho 2006

ÍNDICE DE AUTORES

Índice

Óscar Mealha, 147

Adérito F. Marcos, 123, 213
Alberto José Garcia Ariza, 89
Alexandra Moedas, 39
Alexandra Rentróia-Bonito, 61
Alexandre Mano, 195
Alfredo Ferreira, 225, 229
Ana Catarina Rua, 105
Ana Paula Afonso, 167
Ana Paula Cláudio, 133, 139
Anabela Cardoso da Silva, 35
André Correia, 95
André Martins, 61
Angela Pimentel, 143
António Ferreira, 57
António Nestor Ribeiro, 77
António Raposo Cordeiro, 19

Beatriz Sousa Santos, 143, 147, 217
Bruno Araújo, 233

Cátia Torres, 43
Carlos Duarte, 85, 105
Carlos Ferreira, 217
Carlos Freitas, 13
Celina P. Leão, 205

Daniel Gonçalves, 157, 177
Daniel P. Gonçalves, 177
David Cruz, 43

Eduardo Santos, 105
Elisabete Sousa Cunha, 35
Elizabeth S. Carvalho, 213

Filipe M. Garcia Pereira, 119
Filipe Miguel Tavares, 9
Florin Zamfir, 147
Frank van Huussen, 143

Gonçalo Fontes, 209
Guilherme Francisco, 105

Helder Pinto, 221
Hugo Simões, 85, 105

Ido A. Iurgel, 123

Jan-Willem Baggerman, 143
Janete Faustino, 139

Javier Rodeiro Iglesias, 73
João Brisson Lopes, 19, 27
João Carlos Silva, 65
João Madeiras Pereira, 27
João Pereira, 233
João Robalo, 105
João Saraiva, 65
Joaquim A. Jorge, 61, 115, 119, 157, 177, 225, 229, 233
Joaquim Madeira, 143, 217
José Creissac Campos, 65, 77, 181, 195
José Miguel Sales Dias, 95
José Nunes, 147
José Santos, 229

Larry L. Constantine, 5
Leonor Moniz Pereira, 19
Luís Alexandre, 209
Luís Bilo, 177
Luís Bruno, 27, 39
Luís Carriço, 43, 53, 85, 105
Luís Carvalho, 11
Luís Duarte, 43, 53
Luís G. M. Ferreira, 81
Luís Garcia, 27, 39, 209
Luís Marcelino, 191

Manuel J. Fonseca, 119, 177, 225, 229
Marco de Sá, 43, 53
Maria Beatriz Carmo, 133, 139
Maria João Varanda, 201
Mariana Cerejo, 229
Mario Marcelo Berón, 201
Michael D. Harrison, 3

Nelson Cavaco, 209
Nuno Barata, 95
Nuno Miguel Eira de Sousa, 181

Pauline van der Horst, 143
Paulo Dias, 143
Paulo Pombinho de Matos, 167
Pedro Antunes, 57
Pedro Arezes, 205
Pedro Nande, 95
Pedro Rangel Henriques, 35, 201
Pedro Semião, 133

Rafael Bidarra, 143
Renata Marin, 157
Roberto Medeiros, 225

Rui Barreto, 225
Rui José, 221
Rui Lopes, 85
Rui Rodrigues, 15

S. F. C. F. Teixeira, 205
Sérgio Neves, 105
Samuel Silva, 217
Sandra Lopes, 35
Susana Gómez-Carnero, 73

Teresa Chambel, 105
Tiago Fernandes, 105
Tiago Guerreiro, 61, 115, 157, 233
Tiago Reis, 53

Vasco Amaral, 191
Vitor Fernandes, 61, 233

ISBN: 972-98464-7-2