



*IV Workshop
sobre Fatores Humanos
em Sistemas Computacionais
Interfaces para Todos*

**De 15 a 17 de outubro de 2001
Florianópolis, SC**

Edição

Walter de Abreu Cybis
Marcelo Soares Pimenta

Promoção

Sociedade Brasileira de Computação (SBC)

Organização

Universidade Federal de Santa Catarina (UFSC)



(Catalogação na fonte pela Biblioteca Universitária
da Universidade Federal de Santa Catarina)

W935a Workshop sobre Fatores Humanos em Sistemas Computacionais (4. : 2001 :
Florianópolis, SC)
[Anais] / Workshop sobre Fatores Humanos em Sistemas Computacionais;
edição Walter de Abreu Cybis [e] Marcelo Soares Pimenta. – Florianópolis :
UFSC, SBC, 2001.
335 p.

ISBN 85-88442-09-4

1. Sistemas homem-máquina – Congressos. 2. Interação homem-máquina –
Congressos. 3. Interfaces (Computador) – Congressos. I. Cybis, Walter de
Abreu. II. Pimenta, Marcelo Soares. III. Título.

CDU: 681.31.011.73

Sumário

Apresentação.....	vi
Comitês.....	viii
Sociedade Brasileira de Computação.....	ix
Palestras, Tutoriais e Mini-Cursos.....	xv

Artigos

<i>A Utilização do Ambiente POIROT como Ferramenta de Apoio ao Método de Avaliação por Comunicabilidade</i> Tatiana. A. Tavares, Jair Leite Cavalcanti.....	1
<i>Acessibilidade na Internet para Deficientes Visuais</i> Andréa dos Santos Rodrigues, Guido Lemos de Souza Filho, José Antônio Borges.....	14
<i>Analisando a Comunicação entre Usuários em Ambientes de Grupo</i> Clarissa Maria de Almeida Barbosa, Raquel Oliveira Prates, Clarisse Sieckenius de Souza.....	25
<i>Analyzing HCI in Multiple Environments: A Case Study</i> Simone Diniz Junqueira Barbosa, Clarisse Sieckenius de Souza, Gilda Dahis.....	36
<i>Avaliação da Navegação de Interfaces Web a partir de Modelos</i> Marco Winckler, Christelle Farenc, Philippe Palanque, Marcelo Pimenta.....	48
<i>Bringing Interface Design and Software Development Processes Together: How Organizational Issues Impact Product's Usability</i> M. Cecília C. Baranauskas, Juliana Salles, Roberto S. Bigonha.....	61
<i>Comparing Usability Evaluation Methods Applied to Corporate Web Portals</i> Cláudia Dias.....	73
<i>Design de Ambientes para EaD: (Re)Significações do Usuário</i> Heloísa Vieira da Rocha, Janne Yukiko Y. Oeiras, Fernanda Maria Pereira Freire, Luciana Alvim Santos Romani.....	84
<i>Design Ergonômico de Software de Correio e Agenda Corporativos na Web</i> Antônio Marcos Endler, Lia Buarque Macedo de Guimarães, Flávio Sanson Fogliatto.....	96
<i>Estudo e Avaliação da Usabilidade de Agentes Improvisacionais de Interface</i> Márcia Cristina Moraes, Ana Carolina Bertoletti, Antônio C. da Rocha Costa.....	108
<i>Gerando Conhecimento Sobre os Homens, Mulheres e Crianças que Usam Computadores: Algumas Contribuições da Psicologia Clínica</i> Ana Maria Nicolaci-da-Costa, Carla Faria Leitão, Daniela Romão-Dias.....	120

<i>Interação com Agentes Pedagógicos Animados: Um Estudo Comparativo</i> Patrícia A. Jaques, Tânia Kist, Evandro Franzen, Marcelo Pimenta, Rosa Vicari.....	132
<i>Layering via Interjeições: Possibilidades de Detalhamento Contextual e Progressivo de Informações de Help</i> Milene Selbach Silveira, Simone Diniz Junqueira Barbosa.....	145
<i>Modelando a Interface de Aplicações em Sistemas de Informação Geográfica como Espaços de Comunicação</i> Juliano Schimiguel, M. Cecília C. Baranauskas, Claudia Bauzer Medeiros.....	157
<i>O Uso de Técnicas de Visualização de Informação como Subsídio à Formação de Comunidades de Aprendizagem em EaD</i> Luciana Alvim Santos Romani, Heloísa Vieira da Rocha.....	169
<i>Personal and Group Spaces: Integrating Resources for Users of Digital Libraries</i> J. Alfredo Sánchez, Carlos Proal, Dámaris Pérez, Ana R. Carballo.....	183
<i>Personalizando a Interação Humano Computador para Buscas em Bibliotecas Digitais</i> Jiani Cordeiro Cardoso, João Batista Oliveira.....	195
<i>Proposta e Validação de Método para Estudo da Percepção de Signos por Sujeitos Inseridos em Diferentes Meios Culturais</i> Adéle Malta Pontes, Afonso Inácio Orth.....	207
<i>Reengenharia de Interfaces com Realidade Virtual Considerando o Paradigma de Orientação a Objetos</i> Cristiano L. Soares, Júnia C. Anacleto Silva, Rosângela D. Penteado.....	217
<i>Semiotic Considerations on Direct Concept Manipulation as a Distinct Interface Style for Learnware</i> Clarisse Sieckenius de Souza, Kamran Sedig.....	229
<i>Um Modelo Conceitual para Programação por Usuários Finais</i> Sérgio Roberto P. da Silva, Clarisse Sieckenius de Souza.....	242
<i>Uso de Interfaces Abundantes em Informação para Exploração Visual de Dados</i> Manoel Gomes de Mendonça Neto, Márcio Oliveira Almeida.....	256
 Posters <hr/>	
<i>A Norma ISO 9241:10 e a Participação do Usuário na Especificação da Usabilidade de Sistemas de Recuperação da Informação (SRI) na Web</i> Maria Esther Russo Lima, Walter de Abreu Cybis.....	271
<i>Alternativas de Interação numa Sociedade que Envelhece</i> André Junqueira Xavier, André Raabe, Márcia Barros de Sales.....	275

<i>Análise Ergonômica do Estado da Arte como Facilitadora do Processo de Especificação de IHC: O Caso de um Ambiente de Autoria em RV</i> Walter de Abreu Cybis, Vera R. N. Schuhmacher, Kelly Cristina Goedert.....	278
<i>Ctrl-P: Enhancing the Readability and Comprehension of Printed Academic Materials</i> Mario Alberto Moreno Rocha.....	281
<i>Desenvolvendo Sistemas Interativos com a UML segundo o Princípio de Independência do Diálogo Humano-Computador</i> Elizabeth Furtado, Régis Simão.....	284
<i>Ergonomia da Informação: Apontamentos para Webdesigners</i> Anna Lúcia dos Santos Vieira e Silva.....	288
<i>GOMSQ: Um método de modelagem de tarefas para sistemas educacionais baseado em comunicabilidade</i> Catharine Ferreira Bach, Raquel Oliveira Prates, Rosa Maria Videira de Figueiredo.....	290
<i>Logo3D - Uma Ferramenta de Auxílio na Aprendizagem de Algoritmos</i> Silvano Maneck Malfatti, Deise Juliana Francisco, Maria Augusta Silveira Netto Nunes, Selan Rodrigues dos Santos.....	294
<i>Structuring the Space of Interaction Practices in the Segmentation of Medical Images: a First Discussion</i> Sílvia D. Olabarriga and Marcelo S. Pimenta.....	297
<i>Sugestões para a Criação de Documentos Web - Visando a Usabilidade</i> Eliane da S. A. Diniz, Cristiane O.Vieira.....	300
<i>Superagui Roleplaying Game: Proposta de uma Ferramenta na Sensibilização para Educação Ambiental</i> Ana Cláudia Bevacqua, Juliana Schwartz, Hilton José da Silva Azevedo, Líbia Patrícia Peralta Agudelo, Marcelo Stein de Lima Sousa.....	304
<i>Um Agente Inteligente Baseado na Engenharia Semiótica Inserido em um Ambiente de Aprendizado à Distância</i> João Jung, Patrícia Jaques, Adja Andrade, Rafael Bordini, Rosa Vicari.....	307
<i>Validação de Recomendações Propostas para Projeto de Interfaces de Aplicações Institucionais de Aprendizagem</i> Marilda S. Chiaramonte e Cora H. F. Pinto Ribeiro.....	310
Índice por Autor	314

Apresentação

É com satisfação que apresentamos os Anais do IHC2001, o IV Workshop sobre Fatores Humanos em Sistemas Computacionais. Eles retratam o encontro da comunidade brasileira de IHC (Interfaces Humano-Computador), ocorrido entre os dias 15 e 17 de outubro de 2001, em Florianópolis, Santa Catarina. Dele participaram profissionais brasileiros, mas também colegas de diferentes nacionalidades, atuando em domínios distintos como semiótica, ergonomia, engenharia de software, design, educação, entre outros. Eles vieram mostrar o que de melhor realizaram em ciência e tecnologia de IHC durante este último ano. De fato, são poucos os encontros de IHC no mundo que são capazes de mostrar tão variada e qualificada produção científica e tecnológica. Estão sendo apresentados 22 bons artigos completos e 14 resumos, selecionados entre as mais de 100 submissões que tivemos este ano. Quem acompanha os IHCs, desde sua primeira edição em 98, em Maringá, passando por Campinas, Gramado e este ano em Florianópolis, pode constatar que crescemos muito em números. Esta afluência recorde de submissões representou para o Comitê de Programa, uma carga de trabalho elevada, mas reconfortante, quando constatada a qualidade de nossa produção. Nosso colega e presidente do CP do IHC2001, Prof. Marcelo Soares Pimenta, foi certamente colocado em uma situação muito difícil, ao ter que dirigir os trabalhos de seleção de apenas 1 entre cada 5 artigos submetidos. Apesar das dificuldades, podemos assegurar que o resultado de todo este processo foi excelente e agradecemos a todas as pessoas nele envolvidas, direta e indiretamente.

Além do desenvolvimento contínuo e acentuado de nossa capacidade científico-tecnológica, os anais do IHC2001 retratam um outro motivo de satisfação muito especial para nossa comunidade. Trata-se da participação de dois dos mais lidos e prestigiados autores em IHC de nossa época, os professores doutores Ben Shneiderman e Jenny Preece. De fato, desde o IHC2000 em Gramado, eles buscavam poder compartilhar conosco suas experiências e perspectivas, que estão agora registradas em nossos anais. Somos gratos aos colegas do CP que muito facilitaram nossos esforços para viabilizar a vinda destes dois autores.

“Interfaces para todos”, o tema do IHC2001 sugere a responsabilidade que temos perante a população brasileira de potenciais usuários da informática. A explosão da Web nos atribui a missão de desenvolver interfaces com acessibilidade, para que a Internet não se torne mais um fator de exclusão social em nosso país. Os anais do IHC2001 registram nossa reflexão sobre este tema, através do resumo do que foi a mesa redonda “Inclusão Digital”, organizada pelo colega Clorsival Pereira Jr. . Eles registram também, a realização do tutorial “Desenvolvimento de interfaces com o usuário de sites Web”, apresentado pelo Prof. Jair Cavalcante Leite, e do mini curso “Avaliação de sites web”, ministrado pelo colega Marco Winckler. Agradecemos aos colegas citados pelas oportunidades de reflexão e atualização que nos proporcionaram.

Enfim, nossos anais atestam que IHC no Brasil não é mais uma novidade, é um fato. Em apenas dois anos ligada a Sociedade Brasileira de Computação, a comunidade de IHC se destaca pelo crescimento e pelo potencial de integração. Desde nossa primeira associação com a comunidade de Engenharia de Software, em 98, temos sempre buscado novos encontros, e neste ano, nos reunimos com colegas da Realidade Virtual, (SRV), da Multimídia (o SBMídia) e da Computação Gráfica (SIBGRAPI). Este encontro conjunto, sem precedentes e

de formidável coesão, denominado INFO2001, foi organizado pelo Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, e temos muito que agradecer à equipe de professores e funcionários do INE, com os quais compartilhamos os esforços de organização dos quatro encontros. Agradeço em particular, ao Comitê de Organização local do IHC, coordenado pela colega Vera Shuhmacher, pela dedicação, responsabilidade e entusiasmo emprestados à tarefa de organizar o nosso encontro.

Finalmente, registramos o apoio decisivo que recebemos do CNPq, da Capes, da Sociedade Brasileira de Computação e da Universidade Federal de Santa Catarina, através de seu Centro Tecnológico, de seu Departamento de Informática e Estatística e de sua Fundação de Ensino de Engenharia do Estado de Santa Catarina.

Obrigado pela atenção e desejo à todos uma boa leitura.

Walter de Abreu Cybis

Coordenador Geral do IHC2001

Comitês

Coordenadores

Walter de Abreu Cybis (INE/UFSC), Coordenador

Vera Rejane Niedersberg Schuhmacher (CPGCC/UFSC), Coordenadora Adjunta

Comitê do Programa

Marcelo Soares Pimenta (UFRGS), Coordenador

Ana Cristina Bicharra Garcia, (UFF)

Ana Maria Nicolaci, (PUC-Rio)

Anamaria de Moraes, (PUC-Rio)

Antonio Carlos dos Santos, (UFSCar)

Clorisval Gomes Pereira Jr., (RNP)

Elizabeth Furtado, (UNIFOR)

Fábio Nogueira de Lucena, (UFG)

Heloisa Vieira da Rocha, (UNICAMP)

Jair Cavalcanti Leite, (UFRN)

Maria Cecilia Calani Barauskas, (UNICAMP)

Leila Gontijo, (EPS-UFSC)

Leland McCleary, (USP)

Lia Buarque Guimarães, (PPGEP-UFRGS)

Marília Levacov, (PPGCOM-UFRGS)

Neri dos Santos, (EPS-UFSC)

Raquel Oliveira Prates, (UERJ)

Renata Vieira, (UNISINOS)

Roberto Cabral de Melo Borges, (II-UFRGS)

Simone Diniz Junqueira Barbosa, (PUC- Rio)

Walter de Abreu Cybis, (LabiUtil-UFSC)

Comissão Organizadora

Antônio Carlos Mariani (UFSC)

Luciano Gamez (UDESC)

Márcia Barros de Sales (CPGCC/UFSC)

Marcelo Morandinni (UEM)

Marta Rosatelli (UFSC)

Mauro Notarnicola Madeira (UNISUL)

Roberto Willrich (UFSC)

Rosamélia Parizzotto (CEFET)

Olinto Varella Furtado (UFSC)

Vitório Bruno Mazzola (UFSC)

Sociedade Brasileira de Computação

Diretoria

Presidente: Flávio Rech Wagner (UFRGS)
Vice-Presidente: Luiz Fernando Gomes Soares (PUC-Rio)
Eventos e Comissões Especiais: Ana Teresa de Castro Martins (UFC)
Administração e Finanças: Taisy Weber (UFRGS)
Educação: Marcos José Santana (USP – São Carlos)
Publicações: Cláudia Maria Bauzer Medeiros (UNICAMP)
Planejamento e Programas Especiais: Robert Carlisle Burnett (PUC-PR)
Secretarias Regionais Aleardo Manacero Jr. (UNESP - São José do Rio Preto)
Divulgação e Marketing: Sérgio Cavalcante (UFPE)
Regulamentação da Profissão: Roberto da Silva Bigonha (UFMG)
Eventos Especiais: Ricardo de Oliveira Anido (UNICAMP)

Conselho – Membros Titulares

Ana Carolina Salgado (UFPE)
Cláudia Bauzer Medeiros (UNICAMP)
Daltro José Nunes (UFRGS)
José Carlos Maldonado (USP/São Carlos)
Paulo Cesar Masiero (USP/São Carlos)
Rosa Maria Vicari (UFRGS)
Sergio de Mello Schneider (UFU)
Silvio Romero de Lemos Meira (UFPE)
Therezinha Souza Costa (PUC-Rio)
Tomasz Kowaltowski (UNICAMP)

Conselho - Membros Suplentes

André Carlos P. de Leon F. de Carvalho (USP/São Carlos)
Itana Maria de Souza Gimenes (UEM)
Jaime Simão Sichman (USP)
Miguel Jonathan (UFRJ)
Raul Sidnei Wazlawick (UFSC)

Comissão de Educação

Ariadne Carvalho (UNICAMP)
Daltro José Nunes (UFRGS)
Fernando Fonseca (UFPE)
José Carlos Maldonado (ICMC-USP)
Maria da Graça Brasil Rocha (UFSCar)
Maria Izabel C. Cabral (UFPB)
Nivaldi Calonego Jr.
Ricardo Augusto da Luz Reis (UFRGS)
Roberto da Silva Bigonha (UFMG)
Therezinha Souza Costa (PUC-Rio)

Secretários Regionais

Regional Santa Catarina: Everaldo Artur Grahl (FURB)
Regional Paraná: Maria Angelica de O. Brunetto (UEL)

Regional Centro Oeste: Maria Elenita M. do Nascimento (UNB)
Regional Minas Gerais: Marcelo Maia (UFOP)
Regional Nordeste 1: Riverson Rios (UFC)
Regional Nordeste 2: Hermano Perrelli de Moura (UFPE)
Regional Nordeste 3: Aline Andrade (UFBA)
Regional Norte: Afonso Jorge Ferreira Cardoso (EMBRAPA)
Regional Rio/Espírito Santo: Paulo Roberto Pinheiro Torres Filho (PUC-RIO)
Regional São Paulo 1: Paulo Centoducatte (UNICAMP)
Regional São Paulo 2: Aleardo Manacero Jr. (UNESP)
Regional Rio Grande do Sul: Luciana Porcher Nedel (UFRGS)

Sócios Institucionais

Instituto Metodista Bennett
Centro Federal de Educação Tecnológica da Paraíba
UNIPAR - Universidade Paranaense
Sociedade Riopretense de Ensino e Educação Limitada
UNICAP - Universidade Católica de Pernambuco
UCS - Universidade de Caxias do Sul
PUC/MG - Pontifícia Universidade Católica de Minas Gerais
FURB - Fundação Univ. Regional de Blumenau
Sociedade Campineira de Educação e Instrução (PUCCamp)
INPE - Instituto Nacional de Pesquisas Espaciais
Fundação Paulista de Tecnologia e Educação
Centro de Ensino Nobel
CEUG - Centro de Ensino Universitário de Guaíra
Sociedade Mineira de Cultura
PUC-Rio - Pontifícia Univ. Católica do Rio de Janeiro
UFMG - Universidade Federal de Minas Gerais
UFRGS - Universidade Federal do Rio Grande do Sul
Sociedade Técnica Educacional de Minas Gerais - Faculdade COTEMIG
Fundação São João Batista
Universidade Estadual de Ponta Grossa
Faculdade Cenecista Presidente Kennedy
Centro de Ensino Superior de São Carlos
UnP - Universidade Potiguar
Sistema COC de Educação e Comunicação S/C LTDA.
Ins. Euro-Americano de Educação, Ciência e Tecnologia
CM Consultoria de Administração S/C LTDA
PUCRS - Pontifícia Univ. Católica do Rio Grande do Sul
Instituto de Ensino Superior THATHI
Universidade Tiradentes
Faculdade Senac de Ciências Exatas e Tecnologia
Fundação de Ensino Eurípides Soares da Rocha
Sociedade de Ensino Superior Toledo S/C LTDA
Instituto Luterano de Ensino Superior de Santarém
Centro Universitário Luterano de Palmas
Instituto Educacional Piracicabano

Fund. Educacional de Criciúma - FUCRI/UNESC
Universidade Católica Dom Bosco
Faculdade de Administração e Informática de Santa Rita do Sapucaí
Faculdade Cecenista de Capivari
Centro de Ensino Sup. de Juiz de Fora
Instituto de Ensino Superior Cecenista
UNISO - Universidade de Sorocaba
Centro Univ. do Leste de Minas Gerais
Centro de Estudos Unificados Bandeirante - CEUBAN
Faculdade de Informática de Taquara
FACTU - Fac. de Ciências e Tecnologia de Unaf
Aurea Administração e Participações S/A
Centro Universitário Positivo
Centro Universitário La Salle - Unilasalle
Faculdade Regional do Vale do Aço - UNIPAC
Centro Universitário Nossa Senhora do Patrocínio
Associação de Ensino de Itapetininga
Centro Universitário Nove de Julho-UNINOVE
Fac. de Tecnologia e Ciências Cons. Lafaiete
Universidade Plesbiteriana Mackenzie
Universidade Católica de Petrópolis
Faculdade de Tecnologia do Vale do Araguaia
Instituto Luterano de Ensino Superior de Manaus - ILES-MAO
Instituto Manaura de Ensino Superior
Faculdade Imperatriz
Universidade de Uberaba
Universidade da Amazônia - UNAMA
Fac. Integradas Hebraico Brasileira Renascença
UNISINOS - Universidade do Vale do Rio dos Sinos
Pontifícia Univ. Católica do Paraná - PUCPR
Faculdades Integradas Espírito Santense
Universidade Iguaçú
Universidade Ibirapuera
Fac. de Processamento de Dados de Joinville
Faculdade Politécnica de Jundiá
Faculdade de Jaguariúna
CEUC - Centro de Ensino Universitário de Cianorte
Fac. Ruy Barbosa de Ciência da Computação
FAFIMAN- Fac. de Fil., Ciências e Letras de Mandaguari
"Sociedade Educacional da Bahia S/C LTDA "
Faculdade de Administração e Informática de Maringá
União de Tecnologia e Escolas de Santa Catarina -UTESC
Centro Reg. Univ. de Espírito Santo do Pinhal
Universidade Gama Filho
Universidade Federal de Rondônia
LNCC - Laboratório Nacional de Computação Científica
Centro de Ensino Superior do Pará



*IV Workshop
sobre Fatores Humanos
em Sistemas Computacionais
Interfaces para Todos*

Palestras, Tutoriais e Mini-Cursos

Palestras Internacionais

Online Communities: Designing Usability, Supporting Sociability

Jenny Preece

Chair & Prof. of Information Systems

University of Maryland Baltimore County

Baltimore, MD 21250

<http://www.ifsm.umbc.edu/~preece/>

Like twentieth-century architects and town planners, online community developers shape digital landscapes, but successful communities must have a clear purpose, people and social policies.

Millions of people meet online to debate baseball scores, compare parenting experiences, get stock information, and check consumer advice. They create communities by their presence or absence, their behavior and personalities. Developers can't control what people do but they can influence them by defining purposes, policies and creating safe, enticing places for people to meet. Knowing how to support social interaction (i.e., sociability) and human-computer interaction (i.e., usability) produces thriving online communities instead of electronic ghost towns.

In this talk I will discuss how developers can create sociability and usability for different kinds of online communities. Compelling examples from research on empathy, hostility and lurking illustrate key points. I also suggest how online communities can influence building social capital.

The Future of the Web: Visual, Social and Universal

Ben Shneiderman

Professor in the Department of Computer Science

University of Maryland

<http://www.cs.umd.edu/users/ben/>

The excitement and fear of the World-Wide Web are both justified. It is a compelling and transforming technology that is just beginning to spread. Its impact will be huge. But it is a technology whose evolution we can shape – if we think about what human values we hold dear, and what kind of a world we wish to create.

This talk will show examples of advanced information visualization technologies visualization tools for financial, commercial, and professional applications are beginning to appear and profoundly expand user capabilities. Social interaction through online communities is sweeping through the World Wide Web and could restructure relationships and governments, but only if trust, privacy, and empathy can be ensured. And finally I will argue for universal usability – bridging the digital divide by facilitating access by the broadest possible set of users.

Tutorial

Desenvolvimento de Interfaces com o Usuário de Sites Web

Prof. Dr. Jair Cavalcanti

Universidade Federal do Rio Grande do Norte, Centro de Ciências Exatas,

Departamento de Informática e Matemática Aplicada.

E-mail: jair@dimap.ufrn.br

A rápida expansão da Internet tem aberto inúmeras novas possibilidades para a implantação de serviços computacionais. A Web é, atualmente, o principal veículo para a prestação destes serviços, permitindo atingir um número cada vez maior e mais diversificados de usuários. A demanda por sistemas computacionais na Web é muito grande e urgente, mas desenvolvedores não podem abrir mão da usabilidade. O aumento da oferta dos sistemas torna o usuário cada vez mais exigente, escolhendo aqueles que melhor resolvam os seus problemas. A interface de usuário é uma parte fundamental no sucesso de um sistema Web, pois é a responsável direta em fazer com que o usuário consiga realizar suas tarefas de maneira fácil, rápida e satisfatória.

Este tutorial tem por objetivo apresentar os conceitos e características de interfaces de usuário em sistemas computacionais para a Web, apresentando os principais estilos de interface, técnicas de interação e tecnologias de desenvolvimento e implementação. Será mostrado como o processo de design da interface pode ser conduzido e quais as principais formas de avaliação da usabilidade.

Conteúdos: Introdução aos Sistemas Web: conceitos, modelos, arquitetura e tecnologias. Interatividade na Web: estilos de interfaces, técnicas de interação e tecnologias de desenvolvimento e implementação. Engenharia de Software para a Web e o Processo de Design de Interfaces. Análise de domínio, análise de usuário e análise de tarefas. Design: Engenharia Semiótica aplicada ao Design de Interfaces. Arquitetura da Informação, Design da Interação e Design Gráfico. Técnicas de Modelagem de Interfaces Web. Avaliação: Técnicas e métricas para avaliação de usabilidade e comunicabilidade de sistemas Web.

Atividades: As atividades do tutorial serão aulas expositivas ilustradas com vários exemplos de casos reais. Serão realizados pequenos exercícios práticos que possibilitem uma aplicação imediata das técnicas apresentadas.

Mini-Curso

Avaliação de Sites Web

Prof. Marco Winckler - Universidade de Sciences Sociales de Toulouse

E-mail: winckler@univ-tlse1.fr

Público-alvo: O muni-curso é aberto a todas as pessoas envolvidas em desenvolvimento Web, designers Web, webmasters, e demais interessados em avaliar a usabilidade de aplicações neste ambiente. Não é especialmente necessária uma formação prévia em IHC mas um conhecimento mínimo sobre o funcionamento do ambiente Web e alguma experiência de desenvolvimento de páginas Web (desenvolvimento de uma página pessoal, por exemplo) são desejáveis.

Objetivos: O objetivo deste mini-curso é apresentar e discutir os principais problemas relacionados à avaliação de usabilidade de sites Web incluindo: limites para realização da avaliação, seleção dos métodos a serem utilizados e transformação dos resultados encontrados em sugestões para o projeto.

Conteúdo: Introdução sobre usabilidade e Web; dificuldades em avaliar usabilidade de interfaces Web; critérios para avaliação de usabilidade; métodos de avaliação: “thinking aloud protocol”; questionários; avaliação heurística; inspeção de regras ergonômicas; análise de logs; outros métodos; custo x benefício da avaliação de usabilidade; quando realizar a avaliação; roteiro para guiar a avaliação de usabilidade; problemas de usabilidade mais frequentes; limites para avaliação de usabilidade na Web; discussão de estudos de caso.

Mesa Redonda

Inclusão Digital

Prof. Clorisval Gomes Pereira Jr. (RNP)

E-mail : cjunior@nc-rj.rnp.br

Nossa sociedade está vivendo um novo paradigma sócio-econômico caracterizado pelo uso crescente de tecnologias de informação e comunicação. Estas tecnologias permitem a codificação e a transmissão de informações e conhecimento em velocidades e quantidades cada vez maiores, beneficiando aquelas comunidades que fazem uso eficiente dessas tecnologias.

No entanto, a competência no uso dessas tecnologias ainda é privilégio de uma minoria. Nota-se um desequilíbrio onde apenas aqueles que fazem uso eficiente das tecnologias de informação poderão obter os frutos e benefícios do novo paradigma econômico. Esse desequilíbrio caracteriza uma divisão digital entre as comunidades mais desenvolvidas e as comunidades marginalizadas. Essa divisão tende a ser ainda mais evidente em países que já apresentam grandes contrastes sociais, como o Brasil.

A necessidade de capacitação e apropriação de conhecimento para o uso e desenvolvimento de tecnologias de informação, é apontada no Relatório de Desenvolvimento Humano (HDR'2001) das Nações Unidas como um fator chave para o desenvolvimento humano da nossa sociedade. Essa necessidade vai de encontro a iniciativas como o Programa Sociedade da Informação e o Programa de Universalização dos Serviços de Telecomunicações do Governo Brasileiro.

O debate sobre a Inclusão Digital aponta para uma busca de esforços e soluções para permitir que as mais diferentes comunidades tenham: acesso à tecnologia; e condições para se capacitar e obter os conhecimentos necessários para utilizar com um mínimo de competência os recursos de tecnologias de informação e comunicação.



*IV Workshop
sobre Fatores Humanos
em Sistemas Computacionais
Interfaces para Todos*

Artigos

A Utilização do Ambiente POIROT como Ferramenta de Apoio ao Método de Avaliação por Comunicabilidade

Tatiana. A. Tavares, Jair Leite Cavalcanti

Departamento de Informática e Matemática Aplicada – DIMAP
Universidade Federal do Rio Grande do Norte – UFRN
Campus Universitário – Lagoa Nova – 59072-970 – Natal - RN
{tati, Jair}@dimap.ufrn.br

Resumo. Este artigo aborda a utilização do *Poirot* (Tavares, 2001) como uma ferramenta para avaliação de interfaces de usuário segundo à avaliação de comunicabilidade. O *Poirot* é fundamentado nos métodos: inspeção formal de usabilidade (Kahl & Prail, 1994) e testes de comunicabilidade (De Souza, 1999; Prates, 2000). Neste artigo, discutimos o módulo do *Poirot* que trata exclusivamente do suporte a avaliação de comunicabilidade. Este módulo é formado, fundamentalmente por duas ferramentas destinadas ao apoio dos testes de comunicabilidade: USInspector e TagPanel.

Palavras Chave: Avaliação de Interfaces de Usuário, Inspeção Formal de Usabilidade, Testes de Comunicabilidade, Ferramentas de Avaliação.

1. Introdução

A avaliação de interfaces de usuário é uma etapa fundamental no desenvolvimento de software e, especialmente, no processo de design de interfaces. Não é por acaso que a etapa de avaliação é abordada por diferentes autores em modelos para o desenvolvimento de sistemas.

Segundo Hartson (1993), a etapa de avaliação pode ser vista como um ponto central do processo de design de software que deve preocupar-se em manter o designer informado sobre como as etapas de desenvolvimento estão sendo executadas. O modelo discutido por Leite (1999) é mais voltado ao design da interface do sistema e destaca a etapa de avaliação como elo fundamental entre as demais etapas de desenvolvimento, servindo para verificação e validação da análise, do design e da prototipação.

A implantação de uma avaliação pressupõe a consideração de alguns fatores básicos, os quais delimitam o escopo e o tipo da avaliação, são eles: objetivo, critérios, estilo, participantes, métodos e técnicas e, por fim, as ferramentas envolvidas na avaliação (Leite, 1999). A partir da definição destes fatores é delineado um determinado tipo de avaliação. Existem diferentes tipos de avaliação, cada qual com características próprias que avaliam objetivos específicos e implementam métodos de avaliação distintos.

Embora tenha sua importância reconhecida na literatura, a implantação da avaliação necessita de mão-de-obra especializada, ferramentas adequadas, tempo e, conseqüentemente, custos o que torna a adoção dessa iniciativa um tanto resistente na prática de desenvolvimento de software. No entanto, através da avaliação é possível identificar e solucionar erros de interface antes mesmo de implementá-la e com isso, a manutenção tornar-se-á menos dispendiosa, pois os usuários não serão mais os detectores

dos erros. Esse fato, além das conseqüências imediatas de manutenção, é um artifício estratégico dentro do desenvolvimento de sistemas, pois, em um mercado de software competitivo, onde vários produtos oferecem funcionalidades semelhantes, o diferencial está cada vez mais centrado em detalhes alheios à funcionalidade oferecida por estes produtos.

Atualmente, a utilização tanto de métodos quanto de ferramentas de avaliação no âmbito de interfaces de usuários ainda é um privilégio de projetos capazes de arcar com o investimento necessário. Deste modo, é preciso utilizar novas alternativas tecnológicas que tornem possível a popularização tanto dos métodos quanto das ferramentas de avaliação. Neste sentido, propomos o Poirot (Tavares, 2001), uma alternativa para reduzir os custos de implantação da avaliação, que favorece a equipe de desenvolvimento durante a elaboração e avaliação da interface do sistema ao longo de todo o desenvolvimento.

Neste artigo buscamos desmistificar um pouco a etapa de avaliação de interfaces trazendo uma experiência mais ligada ao lado prático de desenvolvimento de interfaces. Para tanto, exploramos o método de testes de comunicabilidade que também partilha desse objetivo e apresentamos um ambiente computacional de avaliação e, em especial, duas ferramentas específicas voltadas para a comunicabilidade. Dessa forma, pretendemos oferecer uma alternativa prática para quem deseja lançar mão de um método de avaliação de interfaces dentro do desenvolvimento de sistemas.

Este artigo apresenta nas seções 2 e 3 um *overview* sobre alguns aspectos teóricos ligados ao Poirot. A seção 4 ilustra o estado da arte tratando-se de ferramentas de avaliação de interfaces. A seção 5 apresenta algumas considerações básicas sobre o ambiente Poirot. A seção 6 apresenta as ferramentas destinadas à avaliação de comunicabilidade, USInspector e TagPanel e, por fim, a seção 7 sumariza os resultados obtidos nos primeiros testes com essas ferramentas.

2. Usabilidade X Comunicabilidade

Um dos fatores mais importantes na avaliação é a definição dos critérios de avaliação. Atualmente, o critério mais comumente adotado é a usabilidade (Nielsen, 1994). Este critério está presente na maioria dos métodos e é fato comum a todas as ferramentas disponíveis (Diniz, 1996; Lecerof, 1998; Mahajan, 1997; Matias 2000; Tavares, 2000). Porém, a usabilidade ainda não constitui um conceito unânime na literatura. A variância deste conceito é dada através das próprias definições dos autores. Essas definições envolvem alguns aspectos considerados aspectos ou características de usabilidade. Números de erros, aprendizado, facilidade de uso, satisfação subjetiva e performance são alguns desses aspectos.

Um outro critério importante a ser considerado na avaliação é a comunicabilidade (De Souza, 1999). A comunicabilidade de um sistema é a propriedade através da qual o sistema transmite ao usuário de uma forma eficaz e eficiente as intenções e princípios de interação que guiaram o design da interface. Este critério permite avaliar se o designer conseguiu comunicar ao usuário o que ele pode fazer e como usar o sistema. Ela permite mostrar que vários problemas na interação usuário-sistema ocorrem por que o designer não conseguiu transmitir ao usuário, através da própria interface, o modelo conceitual da aplicação.

O critério de comunicabilidade está fundamentado na abordagem da Engenharia Semiótica. A Engenharia Semiótica (De Souza, 1993) utiliza a teoria semiótica para justificar as etapas envolvidas no design de interfaces. Nesta abordagem, as interfaces de usuário são vistas

como artefatos de metacomunicação que utilizam signos para representar mensagens de designers para os usuários da aplicação.

A Figura 1 ilustra o escopo da avaliação por comunicabilidade e usabilidade, que podem ser vistas como métodos complementares uma vez que a usabilidade enfoca a comunicação usuário-sistema e comunicabilidade a comunicação designer-usuário através do sistema. A avaliação de comunicabilidade envolve uma visão mais global do sistema voltada a avaliação da comunicação entre designer, sistema e usuários, considerando, sobretudo, a interação. Além disso, esta abordagem está fundamentada na proposta teórica da Engenharia Semiótica, o que propicia uma séria de novas expectativas e justificativas para os fatores considerados.

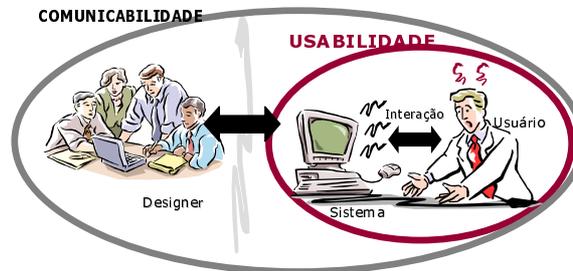


Figura 1: Comunicabilidade X Usabilidade.

3. Método de Avaliação por Testes de Comunicabilidade

O método proposto por Prates, De Souza, e Barbosa (2000) utiliza as idéias da Engenharia Semiótica numa aplicação prática para a avaliação do design de interfaces. Nessa abordagem os sistemas são vistos como mensagens de designers para usuários constituindo uma espécie de conversação. Manter a continuidade nesta conversação significa manter a estabilidade da própria aplicação. O objetivo deste método é a detecção de rupturas na interação. Essas rupturas descrevem situações onde a continuidade na conversação foi rompida ou mal formulada. Desta forma, foi instituído um conjunto de interjeições ou tags capazes de expressar as principais situações de ruptura durante a interação. Em Prates (2000) pode-se a lista completa de tags e seus significados. Alguns exemplos de tags são

- **Que é isso?** O usuário tenta descobrir o que significa um objeto ou ação da interface já que seu significado não lhe parece claro.
- **Para mim está bom...** O usuário obtém um resultado que ele acredita ser o desejado, mas que não o é.
- **Não dá.** O usuário não é capaz de alcançar o objetivo proposto, ou porque os recursos (tempo, paciência, informação desejada, ...) não estavam disponíveis, ou porque ele não sabia como.
- **Socorro.** O usuário não consegue alcançar o seu objetivo e então recorre ao sistema de ajuda (help).

A detecção deste conjunto de interjeições está baseada na identificação de uma série de sintomas que caracterizam cada situação. A associação dessas interjeições com problemas clássicos de interação e usabilidade permite-nos traçar e reutilizar estratégias que

solucionem essa problemática, de modo que, a interface projetada possa atender de forma mais adequada aos usuários do sistema.

Sucintamente pode-se dizer que uma avaliação de comunicabilidade é dividida em duas etapas: a coleta e a análise dos dados coletados. A coleta de dados envolve, basicamente, a determinação de um conjunto de tarefas que serão testadas pelos usuários e a adotar a execução e gravação destas tarefas. Posteriormente, é feita a análise dos dados coletados, conforme está ilustrado na Figura 2 que inclui a realização das seguintes atividades:

- **Tagging** - consiste na identificação das tags (ou interjeições) das interações. Esta etapa pode ser executada por especialistas em semiótica, designers ou pelos próprios usuários.
- **Interpretação** - abrange o mapeamento das interjeições em problemas clássicos de usabilidade e interação ou guidelines de design, por exemplo. Esta etapa deve ser realizada por especialistas que tenham conhecimento dos problemas tratados.
- **Semiotic Profiling** - é o traçado do perfil semiótico (semiotic profiling) que deve ficar a cargo de especialistas em Engenharia Semiótica que analisam os resultados obtidos em termos semióticos

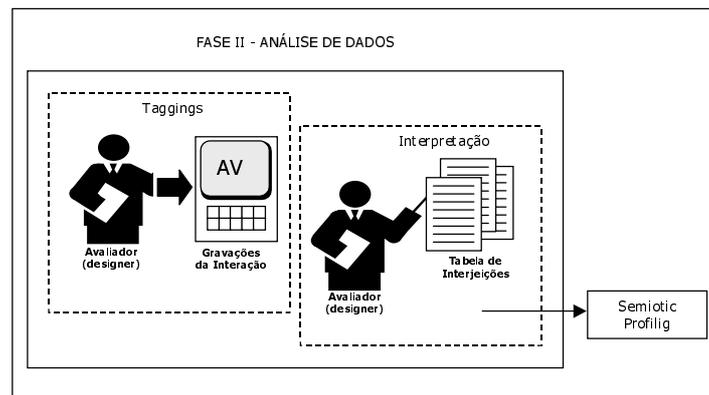


Figura 2: Método de Avaliação por Comunicabilidade –Fase II.

4. Trabalhos Correlatos

Existem diversas ferramentas de avaliação as quais trabalham com critérios e métodos de avaliação distintos. Um dos passos rumo ao desenvolvimento do Poirot foi fazer um levantamento sobre as ferramentas de avaliação disponíveis. Nesta seção procuramos sintetizar os principais pontos desse levantamento destacando algumas vantagens e desvantagens detectadas nessas ferramentas. A Tabela 1 apresenta as comparações entre as ferramentas estudadas, enfatizando, o critério, as técnicas e métodos, o ponto forte e o ponto fraco de cada ferramenta.

Dentre as características comuns entre as ferramentas merece destaque o critério da avaliação. Em alguns casos, como a MetriStation, a Sherlock e a Q.A.S.U. este critério apresenta-se bem específico, focalizando performance do usuário, consistência visual e textual de GUIs e satisfação subjetiva do usuário, respectivamente. No entanto, todas as

ferramentas objetivam a avaliação de usabilidade ainda que de forma parcial. Outro ponto interessante é que embora partilhem do mesmo critério de avaliação, a usabilidade, elas baseiam seu funcionamento em métodos e técnicas de avaliação diferentes. De acordo com o método escolhido essas ferramentas tratam a usabilidade sob diferentes enfoques: (1) Monitoramento Externo - MetriStation monitora e observa todas ações dos usuários através de câmeras de vídeo e gravadores de som. (2) Monitoramento Interno –SPY X SPY registra as ações do usuário através dos eventos gerados pelos arquivos de log do sistema operacional. (3) Opinião do Usuário – QASU registra a opinião do usuário através de questionamentos sobre a interação com a interface.

	Critério	Técnicas/Métodos	Ponto +	Ponto -
MetriStation (Maxion, 1997)	Performance do usuário.	Monitoração e observação	Captura e análise de dados automática.	Necessita de configuração de hardware mínima.
Sherlock (Mahajan, 1997)	Consistência Visual e Textual	Inspeção/Checkagem de padrões.	Automatiza o processo de inspeção.	Necessita de uma entrada padrão
FaiWin (Diniz, 1996)	Layout de Telas	Monitoração.	Captura dados automaticamente.	Restrita a Plataforma de Desenvolvimento
QASU (Tavares, 2000)	Satisfação Subjetiva do Usuário.	Inspeção, Questionários e Avaliação Remota	Fácil de usar – Ferramenta em ambiente Web.	Depende excessivamente do usuário
SPYxSPY (Matias, 2000)	Usabilidade Performance do usuário.	Inspeção – Monitoramento.	Registrar todas as ações dos usuários.	Depende do SO utilizado.

Tabela 1: Comparações entre as ferramentas de avaliação.

Um dos diferenciais do Poirot, apresentado a seguir, é propor ferramentas especialmente projetadas para a avaliação do critério comunicabilidade.

5. O Ambiente Poirot

O Poirot é uma junção de ferramentas computacionais que executam funções estratégicas na avaliação de interfaces. Essas ferramentas estão divididas em quatro grupos: configuração, comunicação, inspeção e suporte. Como o ambiente Poirot provê dois módulos de avaliação: usabilidade e comunicabilidade, para cada módulo de avaliação existem ferramentas específicas e outras comuns.

Uma das grandes vantagens do Poirot é atuar junto aos aspectos gerenciais ligados a etapa de avaliação. Para tanto, o Poirot baseia-se no método de Inspeção Formal de Usabilidade de Kahn e Prail (1994) que tem por objetivo de auxiliar os desenvolvedores na revisão de um produto. Para tanto, o método especifica um processo formal para detecção e descrição de erros de usabilidade, um conjunto de responsabilidades para cada participante da avaliação, e por fim, um framework que define estruturalmente todas as etapas do processo. Deste modo, a sistemática externa é baseada na inspeção formal de usabilidade, isto é, implementa as etapas de gerenciamento da avaliação previstas pelo método (planejamento, reunião, revisão individual, sessão coletiva, design e avaliação). Dentro de cada etapa existem atividades que devem ser executadas. No módulo de comunicabilidade são encaixadas as atividades dos testes de comunicabilidade, o que define uma sistemática interna de trabalho.

A Figura 3 ilustra a tela para submissão de um projeto no ambiente Poirot. Uma vez que o projeto é aceito pelo Poirot devem ser executadas as demais ferramentas de configuração que ajustam o escopo da avaliação solicitada as etapas de execução do Poirot. As configurações definem as atividades de execução do processo de avaliação, o material e as

ferramentas disponíveis, os usuários envolvidos, as tarefas que serão analisadas e o cronograma de trabalho.

The screenshot shows a web-based form for submitting a project to the Poirt toolkit. The form is organized into several sections:

- Project Information:** Includes a field for 'Enter the name for your project in Poirt' (filled with 'Portal Senado') and a 'Choose the evaluation module' section with radio buttons for 'Communicability' (selected) and 'Usability'.
- Personal Profile:** Fields for 'User' (filled with 'Tali'), 'Login', 'Full name' (filled with 'Talana Aires Tavares'), 'E-mail' (filled with 'tal@2@uaap.ufr.br'), 'URL' (filled with 'http://www.senadofederal.gov.br'), 'City' (filled with 'Natal'), 'UF' (filled with 'RN'), 'State' (filled with 'Brasil'), 'Occupation' (filled with 'Engenharia de Software'), and 'Professional Course' (filled with 'Bacharelado em Informática').
- Product Profile:** Fields for 'Name' (filled with 'Site Consultivo do Senado'), 'Description' (filled with 'Site de informação corporativa do Senado Federal'), 'Application Style' (filled with 'Web'), 'URL' (filled with 'http://www.senadofederal.gov.br'), and 'Prototype Mode' (radio buttons for 'Executable' and 'Web Site').
- Development Team:** Fields for 'Size' (radio buttons for 'Less than 5 people', 'Between 5 and 15 people', 'More than 15 people'), 'Phase' (filled with 'Análise'), and 'Method' (filled with 'None').

Buttons for 'Cancel' and 'Submit' are located at the bottom right of the form.

Figura 3: Tela para Submissão de um Projeto.

As ferramentas de comunicação são destinadas ao envio e recebimento de mensagens de forma síncrona e assíncrona entre os usuários. Essa comunicação pode acontecer via correio eletrônico, lista de discussão, mecanismos de chat ou videoconferência. O conjunto dessas ferramentas implementa o que chamamos de sessão virtual de avaliação, onde os avaliadores discutem os problemas detectados e as soluções sugeridas.

As ferramentas de suporte auxiliam as atividades que antecedem a avaliação. O Editor de Modelo de Usuários e o Editor de Cenários de Tarefas já possuem versões funcionais, enquanto que o Repositório de Casos e o Editor de Modelo de Tarefas estão em versão de apresentação. Os softwares de Captura utilizados, popularmente conhecidos como *screencams*, são softwares comerciais.

As ferramentas de inspeção constituem o núcleo funcional da avaliação. Através destas ferramentas o Poirt trata de forma distinta os critérios de avaliação que aproveitam a mesma infra-estrutura do ambiente para avaliar diferentes enfoques. Para tanto, o módulo de usabilidade conta com a ferramenta GEU (Grade de Erros de Usabilidade) e o módulo de comunicabilidade das ferramentas USInspector e TagPanel.

A metodologia adotada no desenvolvimento do Poirt utilizou os paradigmas de prototipação e evolutivo incremental. Após a análise de requisitos, a foi feita a modelagem que adotou como notação a UML (Booch, 2000) e como ferramenta, o Rose. Quanto à estratégia de implementação o Poirt está sendo desenvolvido em fases e utiliza como ambiente de desenvolvimento o Delphi e sua linguagem nativa, o Object Pascal. Atualmente o Poirt conta com protótipos funcionais para as ferramentas ligadas ao módulo de comunicabilidade e com protótipos de apresentação para o módulo de usabilidade e ferramentas de comunicação.

6. USInspector e TagPanel – Ferramentas de Suporte a Comunicabilidade

As ferramentas Poirt destinadas à avaliação de comunicabilidade atuam em duas frentes: junto aos usuários durante os testes (coleta de dados) e junto aos designers durante a análise dos resultados obtidos (análise de dados).

A ferramenta destinada a coleta de dados é o USInspector (*User Inspector*). A função do USInspector é apresentar ao usuário as tarefas que ele deve executar e acionar sua execução e gravação. Paralelamente, o USInspector ainda disponibiliza um módulo para capturar as interjeições detectadas pelo usuário durante a interação, este módulo repassa os resultados obtidos para a ferramenta de auxílio a análise dos dados.

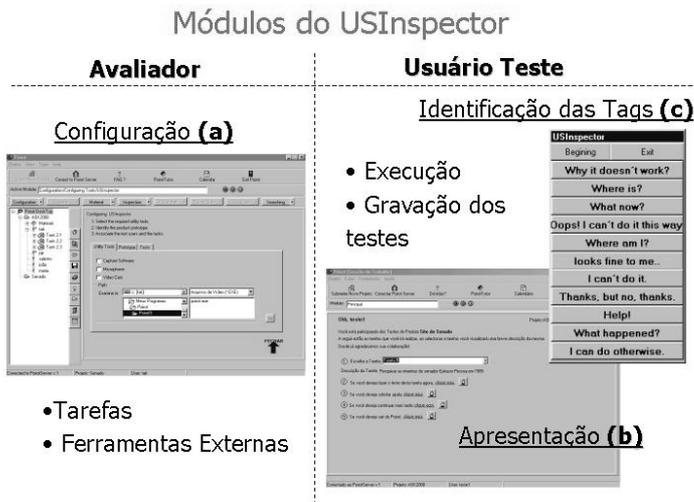


Figura 4: USInspector. (a) Tela de Configuração. (b) Tela de Apresentação. (c) Tela de Identificação das Tags.

A Figura 4 a apresenta a interface para configuração do USInspector. Através desta interface o coordenador deve identificar quais usuários (da categoria usuários-teste) irão executar as tarefas a serem analisadas. Além disso, é necessário especificar a localização de captura e do protótipo. Este protótipo pode ser um executável ou um site na internet, por exemplo.

Quando o USInspector é executado, tanto a gravação quanto o acionamento do protótipo são acionados paralelamente. Através do USInspector, os usuários identificam as interjeições, produzindo um esquema que associa o tempo em que aconteceu a detecção e a interjeição detectada. Esse esquema, então, será incorporado às características da tarefa do projeto Poirot correspondente.

Ao se logar no ambiente o Poirot, um usuário-teste é recepcionado pela tela de apresentação do USInspector (Figura 4-b) que apresenta as tarefas destinadas para ele, assim como, uma breve descrição do processo, iniciando a gravação da interação quando solicitado.

Durante os testes com a aplicação em avaliação é apresentada a tela de interação com o usuário do USInspector (Figura 4-c). Através desta tela, o usuário observa o conjunto de interjeições disponíveis, clicando em uma dessas opções quando enfrenta alguma dessas situações na realização de determinada tarefa.

Além dessas facilidades o USInspector poderá incorporar outros recursos que implementam outras funcionalidades úteis ao acompanhamento dos testes. Estes recursos são

componentes de software e/ou de hardware que têm seu acionamento controlado pelo USInspector. Desta forma, é possível implantar a técnica de *think aloud*, por exemplo, incorporando um gravador de áudio e um microfone, ou ainda, prover mecanismos de observação remota dos testes através de um software para gravação e uma câmera de vídeo.

Enquanto o USInspector é dedicado aos usuários teste, os mecanismos de suporte a fase de análise são indicados para uso de designers e especialistas e são chamados de instrumentos de inspeção. Esses instrumentos funcionam analogamente a um prontuário médico, onde a figura do médico equivale aos designers e especialistas e a interface analisada à paciente. Os instrumentos de inspeção auxiliam designers e especialistas na identificação e descrição dos problemas da interface, através da identificação das tags, seus sintomas e características. As sugestões de solução também são anotadas.

Atualmente trabalhamos com três possibilidades, duas destinadas à comunicabilidade. Uma opção para inspeção individual e outra para coletiva. Além disso, na tentativa de ajudar os designers e especialistas conforme as etapas sugeridas em (Prates, 2000) foram projetados mecanismos que apoiem as etapas de tagging, interpretação e perfil semiótico. Estes mecanismos são baseados nos resultados dos instrumentos de inspeção e tem seu objetivo centrado na discussão desses resultados.

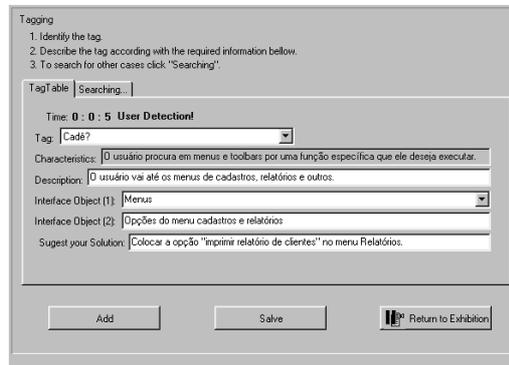
O instrumento de inspeção individual dispara a exibição dos vídeos das interações e possibilita que anotações sejam feitas ao longo desta apresentação. Como foi visto, essas anotações são referentes as tags detectadas e as sugestões para corrigi-las. O esquema proveniente do USInspector é processado simultaneamente com a exibição da gravação para que o inspetor possa perceber sempre que o usuário tenha detectado alguma tag e possa analisá-la. A inspeção individual é feita através de um exibidor do vídeo com a gravação da interação que disponibiliza os seguintes comandos de controle: iniciar, pausar e parar. O comando de pausa apresenta uma particularidade, isto é, também é responsável por acionar o mecanismo de identificação das tags. Uma barra de status localizada na parte inferior da janela de exibição destaca as interjeições detectadas pelo usuário a medida em que elas vão acontecendo e a contagem do tempo de exibição.

O módulo de identificação de tags é chamado de *TagPanel*, e é responsável por efetuar as anotações dos inspetores. O *TagPanel* é disparado sempre que o vídeo é pausado. A Figura 5 apresenta a tela de *TagPanel* para inspeção individual. Nesta tela, designers e especialistas devem num primeiro momento utilizar a aba de “TagPanel” para identificar e caracterizar as tags.

As tags são caracterizadas pelos sintomas que também servem de dicas durante a identificação e pela descrição, onde a situação particular correspondente a cada tag deve ser especificada. Além disso, os objetos de interface relacionados com o problema devem ser descritos. Estes objetos são representados em dois níveis: um genérico - Objetos de Interface com Problema (1), que tem por objetivo identificar a classe do objeto, e outro específico - Objetos de Interface com Problema (2) que denota o objeto que apresenta o problema na situação analisada. A seguir são descritas as sugestões de soluções para que o designer ou o profissional responsável pelo desenvolvimento da interface possa efetuar alterações no produto.

Por fim, a aba de “Searching...” provê um atalho para o material de inspeção do produto analisado caso o inspetor necessite de algum modelo disponível para compor suas anotações. O botão “Return to Exhibition” retorna a exibição da gravação da interação.

A inspeção (individual ou coletiva) garante a primeira saída de resultados da avaliação, essa saída, é denominada mapa de interação. Esse mapa corresponde às informações obtidas em sua forma bruta, organizadas por tarefa. Cada tarefa possui um esquema indexado pela variável tempo de exibição que expõe as tags detectadas, seus sintomas, os objetos de interface envolvidos e possíveis soluções. Além disso, é explicitada a origem de detecção das tags, isto é, proveniente dos testes com usuários (USInspector) ou dos próprios inspetores.



The screenshot shows a 'Tagging' window with the following content:

- Instructions: 1. Identify the tag, 2. Describe the tag according with the required information below, 3. To search for other cases click: "Searching".
- Search bar: TagTable Searching
- Time: 0 : 0 : 5 User Detection!
- Tag: Cad?
- Characteristics: O usuário procura em menus e toolbars por uma função específica que ele deseja executar.
- Description: O usuário vai até os menus de cadastros, relatórios e outros.
- Interface Object (1): Menus
- Interface Object (2): Opções do menu cadastros e relatórios
- Suggest your Solution: Colocar a opção "imprimir relatório de clientes" no menu Relatórios.
- Buttons: Add, Save, Return to Exhibition

Figura 5: TagPanel.

Os resultados são construídos por etapas. A primeira análise é destinada a validação do mapa da interação, principalmente, das interjeições detectadas. Nesta etapa é feita a verificação da etapa de tagging através da discussão dos dados obtidos por todos os envolvidos, inclusive pelos usuários teste. Para realização desta etapa através do Poirot, são utilizadas as ferramentas de comunicação síncronas (chat) e assíncronas (lista de discussão do grupo). A Figura 6 ilustra o protótipo para a sessão de avaliação virtual sediada no Poirot. Nesse protótipo utilizou-se uma ferramenta para chat como meio de discussão. A área de trabalho Poirot é dividida em quatro subáreas: (1) Mapa da Interação – área destinada à visualização as informações provenientes do mapa de interação que são agrupadas por tarefa e indexadas pela variável tempo. (2) Área de discussão – área destinada as trocas de mensagens entre os participantes da sessão. (3) Modificações – área para que os participantes enviem suas modificações para o mapa de interação atual. Nesta área, o participante define uma unidade de tempo, o local do problema e a alteração proposta e, então, submete ao coordenador da sessão que fará a modificação no mapa. (4) Área de visualização gráfica dos participantes ativos na sessão.

Ao final desta etapa de discussão, o grupo de avaliação deve chegar a um conjunto de sugestões de alterações para interface do produto, que deve ser passado a equipe de designers responsáveis pelo produto. Esse segundo nível de resultados é reorganizado em função dos objetos de interface que apresentam problemas.

A etapa seguinte é destinada aos designers e especialistas e deve checar e validar as soluções sugeridas com base nos problemas detectados. Então, a etapa de interpretação consiste no mapeamento desses dados para problemas comuns de usabilidade que muitas vezes possuem soluções já conhecidas, principalmente pelos designers. Esta etapa implica na elaboração de um outro nível de resultados destinados aos designers que farão as devidas alterações no produto analisado.

O perfil semiótico é traçado num último nível de análise dos resultados. Este perfil envolve especialistas em semiótica e IHC que analisam os resultados enfocando as razões semióticas que justificam os fenômenos detectados. Esta etapa procura justificar a problemática encontrada e buscar nessas causas as possíveis soluções que não só solucionem esse problema específico, mas classes de problemas dessa ordem.

Desta forma, podemos dizer que os frutos de uma inspeção caracterizam resultados com propósitos diferentes, dedicados a detecção, solução e discussão de problemas de comunicabilidade. Esses resultados são agrupados e armazenados numa base de dados que serve de instrumento de consulta para os inspetores em projetos posteriores.

Essa base de dados, além da aplicabilidade como repositório de informações para consulta pode ser processada por ferramentas especializadas e prover informações de análise ainda mais profunda. Este ponto de vista se fundamenta na teoria baseada em casos de inteligência artificial. Nesta abordagem bases de dados são processadas na busca de regras comuns que possibilitem a extração de conhecimento, isto é, de meta informação sobre os dados armazenados.

Nas versões atuais as ferramentas USInspector e TagPanel já foram submetidas a testes em situações reais de uso, bem como, a utilização de algumas ferramentas de configuração e de suporte (Editor de Modelo de Usuários e Editor de Cenários).



Figura 6: Sessão de Avaliação através do Poirot.

7. Resultados Obtidos

Os primeiros resultados obtidos foram a partir da execução de alguns testes com os protótipos do Poirot. Através destes testes foi possível avaliar e validar nossos objetivos com o desenvolvimento deste trabalho. Duas aplicações foram utilizadas para os testes: uma aplicação convencional *stand alone* (Cechin, 1997) e um *website* corporativo (Dias, 2001). Além disso, foi envolvido um grupo de quatro voluntários que auxiliaram na execução dos testes.

A ferramenta USInspector na versão testada e com a configuração utilizada não se mostrou autônoma para realização dos testes. Os problemas detectados estão relacionados principalmente com o software de captura utilizado. Entretanto, a integração com outros softwares de captura está sendo testada.

A principal vantagem do USInspector é prover um mecanismo de captura de interjeições sob um ponto de vista que não era explorado inicialmente: o usuário. Dessa forma, o usuário colabora na construção do primeiro mapa da interação e fornece um novo recurso para ser explorado pelo avaliador. Porém, muitas coisas ainda devem ser alteradas para dar maior fluência e independência ao USInspector.

Outra facilidade é a possibilidade de incorporar além da gravação da interação outras formas de registro da interação como uma microcâmera ou um microfone. Facilitando, inclusive o envolvimento de outras técnicas como a técnica de *thinking aloud*

Por fim, a utilização do Poirot impõe a produção de resultados em diferentes níveis e direcionados a pessoas que atuam de forma diferenciada no processo de desenvolvimento. Esse fato torna os resultados do Poirot mais efetivos que outros, quando aplicados à correção dos problemas detectados e ao entendimento das situações que levaram a esses erros, isto é, todo problema levantado pela avaliação do Poirot, possui uma descrição do problema e outra da solução. A descrição da solução está sempre atrelada a objetos da interface analisada, o que facilita o processo de correção.

A ferramenta TagPanel mostrou-se capaz de auxiliar o avaliador na identificação e, sobretudo, na descrição das interjeições. Em geral, os resultados de uma avaliação enfocam os erros e não as correções. A possibilidade inovadora de descrever os objetos de interface relacionados com as interjeições foi um salto em direção a correção dos erros levantados pela avaliação e de como e onde corrigi-los. As principais vantagens dessa ferramenta foram:

- A descrição dos objetos de interface 2 (referente aos objetos de interface relacionados com o problema) poderia ser como a descrição 1, isto é, pré-determinada. Isso deixaria mais ágil e uniforme a ferramenta.
- A parte de identificação/descrição das interjeições poderia ser integrada a exibição do vídeo.
- A ferramenta poderia chamar mais atenção quando há interjeições denotadas pelo usuário (isso na tela de exibição) pois às vezes essa informação pode passar despercebida pelo inspetor.

O Poirot foi desenvolvido com o intuito de propor uma alternativa para apoiar o design de software. Dentro desta perspectiva, propor um ambiente de avaliação de interfaces sob diferentes aspectos é uma forma de contribuir na melhora da usabilidade e comunicabilidade dos sistemas. Neste sentido acreditamos que o Poirot atingiu nossas expectativas e abriu novos caminhos para estudo. As ferramentas *USInspector* e *TagPanel* são inéditas no que tange o suporte a avaliação de comunicabilidade e cumprem com os objetivos esperados. Ainda há muito que ser feito, como, a implementação das ferramentas de comunicação, implementação da versão Web e incorporação de outros critérios de avaliação. Mesmo assim nossos objetivos foram alcançados e espera-se que o Poirot possa contribuir para popularização da avaliação de interfaces de usuários dentro do escopo de desenvolvimento de sistemas.

Referências

BOOCH, Grady; RUMBAUGH, James & JACOBSON, Ivar (2000). **The Unified Modeling Language User Guide**. SILVA, Fábio Freitas (org.) UML- Guia do Usuário. Rio de Janeiro: Editora Campus, 2000, 472 p.

- CECHIN, Leonardo; GELATTI, Paola; NUNES, Isabel e TAVARES, Tatiana (1998). **Sistema para uma Corrida de Simulação**. Pelotas-RS, 1998. Trabalho apresentado à Disciplina de Simulação Discreta. Universidade Federal de Pelotas, Instituto de Física e Matemática, Curso de Informática.
- DE SOUZA, Clarisse Sieckenius (1993). **The Semiotic Engineering of User Interface Languages**. International Journal of Man Machine Studies 39, Academic Press.
- DE SOUZA, Clarisse Sieckenius; PRATES, Raquel Oliveira & BARBOSA, Simone (1999). **A Method for Evaluating Software Communicability**. In: C.J.P. de Lucena (ed.) Inf MCC11/1999. Computer Science Department, PUC-Rio.
- DIAS, Cláudia (2001). **Métodos de Avaliação de Usabilidade no Contexto de Portais Corporativos: um Estudo de Caso**. Brasília-DF, 2001 (data prevista). Dissertação (Mestrado em Ciência da Informação). Universidade de Brasília.
- DINIZ, Eliane da S. A (1996). **Ferramenta para Avaliação de Interfaces em Ambiente Windows, a partir da Monitoração de Dados – FAIWIN**. Campina Grande - PB, 1996. Dissertação (Mestrado em Informática). Universidade Federal da Paraíba, Instituto de Informática.
- KAHL, Michael & PRAIL, Amanda (1994). **Formal Usability Inspection**. NIELSEN, Jakob & MACK, Robert L (ed.). Usability Inspection Methods. New York, NY: John Wiley & Sons, 1994, p. 141-169.
- HARTSON, H. Rex & HIX, Deborah (1993). **Developing User Interfaces: Ensuring Usability Thought Product & Process**. John Wiley & Sons.
- LECEROF, Andreas e PATERNO, Fabio (1998). **Automatic Support for Usability Evaluation**. In: IEEE Transactions on Software Engineering, Vol. 24, No. 10, Outubro. 863-888p.
- LEITE, Jair Cavalcanti (1999). **Notas de Aula – Disciplinas de Engenharia de Software e Projeto de Interfaces de Usuário**. <<http://www.dimap.ufrn.br/~jair/pium>>
- MAHAJAN, Rohit e SHNEIDERMAN, Ben (1997). **Visual and Textual Consistency Checking Tools for Graphical User Interfaces** In: IEEE Transactions on Software Engineering, nVol.23. No11 722-735p.
- MATIAS, Márcio; MASSARO Neto, Sílvio & SANTOS, Néri dos (2000). **Uma Ferramenta de Apoio ao Registro da Interação Humano Computador**. In: 3RD WORKSHOP ON FACTORS IN COMPUTER SYSTEMS: MANY FACES IN INTERFACES, Gramado-RS, Brasil, 2000. Anais do 3RD WORKSHOP ON FACTORS IN COMPUTER SYSTEMS, p. 142-147.
- MAXION, Roy A. e SYME, Philip A (1997). **MetriStation: A Tool for User-Interface Fault Detection**. In: Proceedings of the 27th International Symposium on Fault-Tolerant Computing - FTCS '97
- NIELSEN, Jakob & MACK, Robert L (1994). **Usability Inspection Methods**. New York, NY: John Wiley & Sons, 1994.

- PRATES, Raquel Oliveira, DE SOUZA, Clarisse Sieckenius; & BARBOSA, Simone (2000). **A Method for Evaluating the Communicability of user Interfaces**. In: *ACM Interactions methods & tools*, Jan-Fev 2000.
- TAVARES, Tatiana Aires e DINIZ, Eliane da Silva Alcoforado (2000) . **Uma Ferramenta para Avaliação Qualitativa de Interfaces em Ambiente Web**. In: XXVI CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA, 2000, Cidade do México. Anais do XXVI CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA.
- TAVARES, Tatiana Aires e LEITE, Jair Cavalcanti (2001) **Poirot -Um ambiente para avaliação de interfaces de usuário**. In: XXVII CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA, 2001, Venezuela. Anais do XXVII CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA.

Acessibilidade na Internet para Deficientes Visuais

Andréa dos Santos Rodrigues^{1,2}, Guido Lemos de Souza Filho¹,
José Antônio Borges³

¹ Universidade Federal do Rio Grande do Norte

² Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte
Faculdade de Ciências e Cultura e Extensão do Rio Grande do Norte

³ Núcleo de Computação Eletrônica da UFRJ

andrea_santos@bol.com.br, guido@dimap.ufrn.br, antonio2@nce.ufrj.br

Abstract: *This paper describes the status of Internet access by visually impaired in Brazil. It shows some formats that are adequate for the development of homepages, using accessibility rules. Those rules have been tested in a browser, created specially to be operated by blind people, in the scope of the DOSVOX Project, whose implementation is here presented.*

Resumo: *Neste trabalho descreve-se a situação atual em que se encontra o acesso a Internet para as pessoas portadoras de deficiência visual no Brasil. São propostas formas de desenvolvimento de páginas com o emprego de regras de acessibilidade. As regras propostas foram testadas através do uso de um browser adaptados aos deficientes visuais desenvolvido no contexto do projeto DOSVOX, cuja implementação é aqui descrita.*

1. Introdução

Vivemos na sociedade da informação. Um dos fatores críticos para o sucesso nesta sociedade é o acesso e utilização das tecnologias de informação e comunicação. Estas tecnologias devem, portanto estar disponíveis ao maior número possível de cidadãos evitando-se assim a exclusão social. Neste contexto a Internet tem um papel crucial a desempenhar. Ela tem o potencial de “quebrar” barreiras físicas e espaciais, servindo de suporte a um grande número de atividades possíveis de serem realizadas por portadores de deficiência.

Em toda parte do mundo e em todos os níveis da sociedade há pessoas portadoras de algum tipo de deficiência. Deficiência significa “perda ou limitação de oportunidades de participar da vida comunitária em condições de igualdade com as demais pessoas” [6]. Considera-se pessoa portadora de deficiência aquela que apresenta, em caráter permanente, perdas ou anormalidades de sua estrutura ou função psicológica, fisiológica ou anatômica, que gerem incapacidade para o desempenho de atividades consideradas normais para o ser humano [12]. De acordo com a Organização Mundial da Saúde (OMS), 10 % da população brasileira é composta de pessoas portadores de deficiência, distribuídos como mostra a tabela 1 [5].

As novas tecnologias, sobretudo no campo das telecomunicações, permitem-nos hoje ter acesso a fontes de informações impensáveis há bem pouco tempo. A facilidade, a rapidez e a supressão de barreiras geográficas tornam possível o acesso aos mais diversos canais de conhecimento, lazer e comunicação.

Milhões de pessoas acessam a WWW [7] todos os dias, em busca de informações, entretenimento, trabalho, educação, comunicação e comércio, entre outras atividades. Devido a sua popularidade, muitos negócios e provedores de informação são criados através de *web sites* para apresentar seus produtos e serviços. No entanto, as grandes maiorias desses sites não levam em consideração uma parcela importante da população: os deficientes visuais. No Brasil, de acordo com a Tabela 1, os deficientes visuais chegam à cerca de 750.000, sendo que 150.000 têm perda de visão tão grave que os impede de usar um computador através de qualquer aparato ótico. Segundo Borges [10], desses, cerca de 3.000 tem acesso ao computador e a Internet, a grande maioria através do uso do sistema DOSVOX, desenvolvido na Universidade Federal do Rio de Janeiro.

Tabela 1. Distribuição dos tipos de deficiência na população brasileira.

Deficiência	População	Porcentagem
Deficiência Mental	7.250.000	5%
Deficiência Física	2.900.000	2%
Deficiência Auditiva	2.175.000	1,5%
Deficiência Múltipla	1.450.000	1%
Deficiência Visual	750.000	0,5%
Total	14.500.000	10%

Sem uma tecnologia de acesso adequada, os deficientes visuais podem ficar gravemente limitados quanto à quantidade e a qualidade das informações que podem acessar, o que inibe, ou até mesmo impossibilita que eles utilizem plenamente as potencialidades deste meio de comunicação. Para viabilizar o uso da Internet pelos deficientes visuais poucas iniciativas concretas estão sendo realizadas no país. A maior parte dos trabalhos realizados, em relação ao desenvolvimento de pesquisas e elaboração de softwares para ao acesso a Internet, são feitos em outros países, com contextos diferentes dos nossos, principalmente no que se refere ao idioma, perfil do usuário e recursos financeiros necessários para sua implementação [8, 9, 14]. Estes fatores, na maioria dos casos, tornam inadequado o aproveitamento dos sistemas desenvolvidos no exterior.

Segundo Borges[10], os deficientes visuais no Brasil são em sua maioria pessoas semi-analfabetas ou possuem somente a educação básica, com extrema dificuldade de acesso a educação. Estes indivíduos necessitam de uma educação especial adequada às suas necessidades. Neste contexto, a tecnologia da informática dispõe de recursos que possibilitam ao deficiente visual ter melhores condições de acesso à educação e conseqüentemente, possibilita uma melhoria na qualidade de vida, seja através do crescimento intelectual (acesso a informações e educação), pessoal (possibilidade de se comunicar e formas de entretenimento com outros indivíduos em condições de igualdade) e profissional (ter meios adequados para desenvolver uma atividade profissional possibilitando a conquista da independência financeira).

Em relação ao acesso a Internet, a criação de uma "Internet para necessidades especiais" baseia-se na concepção de equipamentos, softwares e conteúdos com características de acessibilidade [11]. Tal facilidade permitiria que pessoas portadoras de deficiência tivessem acesso a um conjunto imenso de informações, estabelecessem contatos e trocassem informações, e encontrassem formas alternativas de lazer, entre outras atividades que o acesso a Internet pode possibilitar.

Alguns países, como Portugal, Estados Unidos, Canadá e Austrália, para facilitar o acesso a pessoas com necessidades especiais regulamentaram a adoção de regras de acessibilidade na concepção da informação disponibilizada na Internet pela administração pública, [7]. Medidas como esta visam divulgar a importância de se criar *sites* acessíveis.

Para auxiliar os desenvolvedores de páginas da *Web* o W3C, organismo responsável pela elaboração de padrões mundiais para a Web, publicou em 5 de Maio de 1999 o primeiro documento: "*Web Content Accessibility Guidelines*". Este documento foi elaborado com o objetivo de propor sugestões de como tornar o conteúdo de documentos web acessível a portadores de deficiência.[4]

Um deficiente visual, ao usar a Internet enfrentará dificuldades para obter as informações apresentadas em formato visual, para interagir com dispositivos diferentes do teclado, para distinguir links e para compreender a estrutura de um documento. A adoção da acessibilidade na confecção das páginas e aplicações para Internet, não se caracteriza como limitação, ao contrário, as regras de acessibilidade tornam os documentos mais flexíveis, rápidos e fáceis de utilizar. O emprego das regras de acessibilidade na confecção de sites permite também a utilização de equipamentos menos convencionais para o acesso a Internet, como por exemplo, o telefone e equipamentos eletrônicos de bolso.

Estas técnicas permitem ainda um aumento na divulgação de páginas e conteúdo em mecanismos de busca como o CADÊ, ALTAVISTA, YAHOO etc.

2. Motivação e Objetivos do Trabalho

Iniciativas de apoio ao deficiente visual no país são escassas. O acesso a Internet se dá de maneira precária, pois além da carência de navegadores apropriados, a forma como as páginas são confeccionadas constitui-se em uma grande barreira, que muitas vezes impossibilita a interação do deficiente visual com as informações e serviços disponíveis na Internet.

Os objetivos deste trabalho foram à realização de estudos sobre a acessibilidade na Internet para deficientes visuais e a elaboração de um browser que possibilita a apresentação em áudio do conteúdo de páginas HTML. O Browser serviu como ferramenta de testes. Buscou-se abordar questões relacionadas com a problemática da acessibilidade, apresentando algumas iniciativas nacionais e internacionais que visam lidar com este problema. Foram propostos alguns princípios baseados nestas iniciativas e em pesquisas e testes realizados com usuários da Internet deficientes visuais. Buscou-se apresentar sugestões de como desenvolver páginas acessíveis. Com este objetivo foram levantados e apresentados os principais erros cometidos no desenvolvimento de páginas que tornam seu conteúdo inacessível aos deficientes visuais.

Como conseqüência deste estudo, espera-se atingir um objetivo bem mais amplo, que é a divulgação de sugestões e orientações de construção de páginas acessíveis aos deficientes visuais, incentivando que desenvolvedores, instituições e empresas divulguem suas informações em formato que possibilite ao deficiente visual pesquisar, estudar, comunicar-se, divertir-se em condições mais próximas as que estão disponíveis para um indivíduo sem deficiência.

3. Acessibilidade

Acessibilidade significa facilidade de interação, aproximação. A acessibilidade no âmbito das tecnologias de informação está associada a ações que tem como objetivo tornar os computadores mais acessíveis a todos os usuários.

Segundo Godinho, [7] a acessibilidade envolve três noções: "Usuários", "Situação" e "Ambiente": O termo "Usuários" significa que nenhum obstáculo deverá ser imposto ao indivíduo face às suas capacidades sensoriais e funcionais. O termo "Situação" significa que o sistema é acessível e utilizável em diversas situações, independentemente do software, comunicações ou equipamentos e o termo "Ambiente" significa que o acesso não é condicionado pelo ambiente físico envolvente, exterior ou interior.

A acessibilidade na Internet caracteriza-se pela flexibilidade da informação e interação. Esta flexibilidade torna possível sua utilização por pessoas com necessidades especiais, bem como a utilização em diferentes ambientes e situações através de vários equipamentos ou navegadores.

Os principais problemas sentidos por usuários deficientes visuais são dificuldades em obter informações apresentadas visualmente; interagir usando um dispositivo diferente do teclado; distinguir rapidamente os links num documento; navegar através de conceitos espaciais; distinguir entre outros sons uma voz produzida por síntese.

O tempo necessário para introduzir técnicas de acessibilidade na concepção de uma página atinge aproximadamente 5% do tempo gasto para escolher uma apresentação visual agradável [13]. As normas de acessibilidade devem ser usadas pelos responsáveis pelo desenvolvimento dos conteúdos, sendo estes responsáveis pela qualidade e quantidade de informações que serão transmitidas aos indivíduos portadores de deficiência.

4. Orientações para a Criação de Sites Acessíveis

O W3C (World Wide Web Consortium) é o órgão que coordena a elaboração e padronização das regras de acessibilidade. Estas regras são adotadas por diversos países e empresas como a IBM e Microsoft [1,2]. As orientações elaboradas pelo W3C tem como objetivo auxiliar e encorajar o desenvolvimento de páginas acessíveis, indicando não só princípios gerais como as formas ideais de implementação que orientam os autores.

De forma genérica os principais princípios para a criação de páginas sejam descritos a seguir:

1. Deve-se utilizar regras padronizadas para que qualquer indivíduo de qualquer parte do mundo tenha possibilidade de criar páginas que são acessíveis universalmente aos diferentes tipos de usuários, independente do tipo de software, hardware e limitação que possua, permitindo a interoperabilidade.

2. As páginas devem prover sempre mecanismos para gerar um texto alternativo quando um elemento não puder ser exibido e devem assegurar que todos os elementos do site são acessíveis via teclado
3. Deve-se utilizar navegação consistente e clara (Evitar o famoso “Clique Aqui!!”), além de colocar informação clara no topo dos cabeçalhos, parágrafos, listas, etc..
4. Outras facilidades importantes são mecanismos para “congelar” as informações que aparecem em movimento. De forma ao sintetizador de voz poderem transformá-las em áudio.
5. Ao invés de destacar alguma informação importante através de cores ou outra forma de formatação utilizando-se elementos visuais deve-se, indicar através de palavras sua importância no contexto da apresentação. *A informação é mais importante do que sua forma de apresentação.*
6. Deve-se criar uma ordem lógica para os links apresentados. Facilitando a navegação. Fornecer links para a página inicial em todas as páginas e garantir que os links textuais são formados por palavras ou sentenças compreensíveis fora do texto.
7. Sempre que se usar elementos gráficos como botões, utilizar texto com a mesma função para facilitar a interação por dispositivos não gráficos e via teclado.
8. Deve-se testar a acessibilidade em diversos browsers, incluindo os browsers com capacidade de sintetizar voz e com leitores de tela e validar com ferramentas de validação como as citadas abaixo:
 - BOBBY: <http://www.cast.bobby.org>
 - W3C HTML Validation Service: <http://validator.w3c.org>

Seguindo estas recomendações os desenvolvedores de páginas estarão abrindo a possibilidade de acesso à informação para a comunidade de deficientes.

5. Síntese de Áudio a partir da interpretação do HTML

Tradicionalmente deficientes visuais podem ter acesso a documentos textos utilizando-se sintetizadores de voz. Entretanto, os deficientes enfrentam problemas quando os documentos possuem outros tipos de elementos, como links, tabelas, formulários, e imagens.

Um browser ideal para estes indivíduos deveria possuir as mesmas características que um browser convencional como: permitir a interatividade, navegar entre documentos através de links, retroceder e avançar entre páginas, gravar, imprimir, copiar, destacar títulos, negritos, sublinhados etc.

O trabalho ora apresentado baseou-se no estudo das estruturas básicas da linguagem HTML, que é o principal elemento na construção de uma página WEB, e de formas de como converter os elementos visuais inseridos na página em som, procurando-se minimizar as perdas de informação neste processo. Esta conversão infelizmente nem sempre é possível, pois ainda não se têm meios de traduzir e interpretar o conteúdo de uma imagem e narrá-la. Este fato motivou a divisão do trabalho em duas partes:

1. Na primeira foram investigadas formas de traduzir comandos do HTML para áudio. Por exemplo, um deficiente visual não tem como saber se determinada palavra simplesmente faz parte do texto ou se é a âncora de um elo. Os sintetizadores de voz simplesmente lêem o texto, sendo necessário haver um tratamento diferenciado para este tipo de objeto.
2. A outra abordagem leva em consideração situações onde não é possível traduzir um elemento da linguagem HTML para áudio. Neste caso, é necessário incentivar os desenvolvedores a criar representações alternativas para os elementos visuais e de formatação. Como, por exemplo, ao invés de colocar uma imagem descrevê-la de forma textual.

Do exposto fica claro que o simples acoplamento de um sintetizador de áudio a um browser convencional não resolve o problema de acessibilidade. A tradução direta de HTML para áudio na maioria dos documentos não funciona adequadamente. Esta constatação motivou a definição da estratégia adotada no trabalho. Optou-se por desenvolver um browser criado especialmente para usuários portadores de deficiência visual. Este browser faz parte de um ambiente operacional também destinado a estes usuários, que será descrito com maiores detalhes posteriormente.

Como contribuição deste trabalho pretende-se possibilitar que um indivíduo portador de deficiência visual seja capaz de acessar a internet e as informações ali contidas de maneira adequada as suas necessidades.

6. WEBVOX

O WEBVOX é um browser que utiliza os módulos de sintetização de voz criados para o DOSVOX (versão para Windows). Como em um browser convencional, o WEBVOX pode tanto carregar uma página da Internet, ou carregar uma página que já esteja disponível localmente, entre outras opções que foram implementadas por outros autores.

Uma vez a página carregada, o módulo WEBTRAD realiza o processamento de seu conteúdo, isto é, a tradução dos comandos HTML para um script de áudio. O *script* texto gerado é "lido" por um outro módulo denominado WEBLEIT que controla o funcionamento do sintetizador de voz.

Uma das contribuições deste trabalho foi o desenvolvimento e aperfeiçoamento de um dos módulos do Browser WEBVOX, o WEBTRAD que faz parte do sistema Dosvox desenvolvido na UFRJ [31]. Este módulo é responsável pela captura do conteúdo de uma página Web e sua transformação para um script de áudio, que poderá ser lido através de um sintetizador de voz. Sua importância concentra-se no fato de que um deficiente visual, não pode ter acesso a informações que estão em formato gráfico, pois os sintetizadores de voz são incapazes de processar tais informações. O módulo denominado WEBTRAD processa páginas HTML, "filtra" as informações textuais e transforma alguns comandos de formatação gráfica em texto ou explicação textual. Sua implementação é semelhante àquela usada em softwares como WAB [8] e PwWebspeak [14].

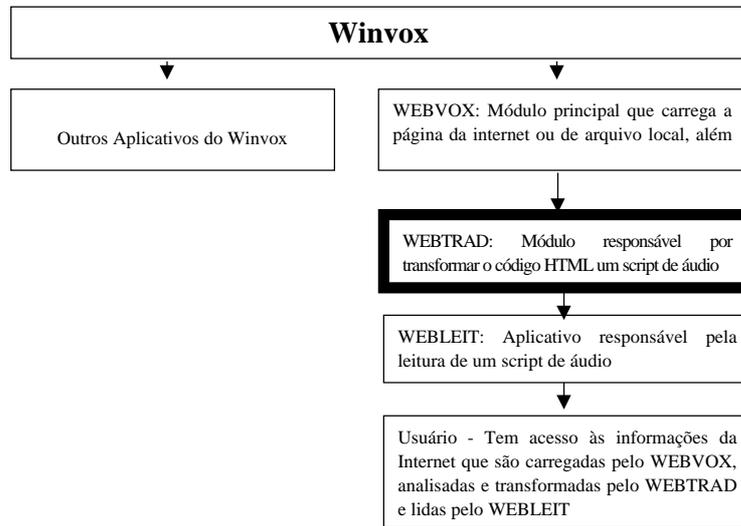


Figura 1 - Resumo do Funcionamento do WEBVOX

Infelizmente, o acesso dos deficientes às páginas na Web não depende apenas do bom funcionamento do protótipo, ou seja, das transformações que ele é capaz de realizar. Depende também da forma como as páginas são desenvolvidas. Sendo assim, outra contribuição importante deste trabalho foi o estudo e aplicação das regras de acessibilidade apresentado em seção anterior. Sem as regras de acessibilidade, muitas vezes o browser é incapaz de apresentar as informações das páginas, principalmente nos casos onde que a página é composta por muitos elementos gráficos.

Maiores detalhes sobre o protótipo implementado são apresentados nas seções que seguem.

7. O WEBTRAD

O WEBTRAD toma o texto HTML e o interpreta, comando a comando, gerando um código intermediário. Embora quase todos os *tags* HTML sejam interpretados (ou pelo menos analisados), um subconjunto destes é de fundamental importância na geração do áudio. Um resumo desses comandos HTML, e como eles são tratados no módulo WEBTRAD é apresentado na tabela 2.

Os comandos implementados no protótipo garantem a navegação básica dentro de uma página padrão que contém links, imagens e texto, indicando a presença de links e permitindo ao usuário, depois da leitura do texto, ler somente a lista de links. A Figura 2 apresenta um exemplo de código HTML. A apresentação em um browser convencional é mostrado na Figura 3. O script de áudio correspondente é mostrado na Figura 4. Estas figuras ilustram o processo de tradução do WEBTRAD.

Tabela 2: Comandos HTML tratados no WEBTRAD

Comando	Tradução
TITLE	Apresenta-se em áudio o título da página, informando ao usuário a página em que está no momento, facilitando assim a navegação.
BR	Indica-se de forma sonora a quebra de uma linha
P	Indica-se de forma sonora o início de um parágrafo
IMG	Se a imagem não possui um texto alternativo, indica-se somente que naquela parte do texto existe uma imagem que não pode ser traduzida.
ALT	Atributo da tag IMG. Se a imagem possui um texto alternativo este texto é exibido em seu lugar.
HR	Indica que existe uma linha horizontal separando seções do documento, as linhas, que são imagens, são substituídas por um sinal sonoro.
OL, UL...	Estes comandos, indicativos de listas (tabelas) são muito importantes, pois a tabela representa mais do que uma disposição visual, mas também uma semântica que inter-relaciona os textos mostrados dentro dela.
HREF	Indica-se a presença de um elo através de sinais sonoros delimitando o início e o fim da âncora e separa-se o endereço para posterior navegação.

Esta seção prossegue com discussões mais detalhadas sobre diversos elementos HTML são tratados.

Acentuação: Como o conteúdo da página é lido pelo sintetizador através do módulo LEVOX, os comandos de acentuação do HTML precisam ser traduzidos para o formato de entrada no sintetizador. Por exemplo, o código HTML “caf&ecute;” é traduzido e incluído no script de áudio na forma da palavra “café”.

Título Principal e Títulos e Subtítulos da Página: O título principal da página é apresentado de forma destacada para facilitar a identificação da página na qual o usuário se encontra. Indica-se em áudio a página que está sendo acessada, através da leitura do título principal da página, a semelhança do que outros browsers convencionais fazem apresentando o título da página em destaque na barra de títulos.

Comandos de Formatação: Os comandos de formatação do HTML são importantes para destacar elementos do texto e estruturar a informação a ser apresentada. Na tradução realizada para áudio são associados sons a estas formatações (texto centralizado, negrito, etc), possibilitando ao usuário, com um treinamento específico, identificar os elementos da página.

Imagens: Quando é encontrada uma Tag indicando a presença de uma figura na página, o tradutor procede da seguinte forma: se a figura não possuir um texto explicativo, o browser simplesmente indica a presença de figura emitindo um sinal sonoro. Quando a figura possui um texto explicativo este é exibido em seu lugar. Note que se a imagem não possui um texto alternativo, indica-se para o usuário que naquela parte do texto existe uma imagem que não pode ser traduzida. Mesmo ocorrendo uma quebra na transmissão das informações, o usuário ao menos é informado da existência da figura. Adicionalmente é

gerada uma mensagem ao desenvolvedor, caso exista e-mail definido na página, indicando o problema de acessibilidade.

Elos (Links): Como pode ser observado na figura 3; em um browser convencional os links são destacados por uma linha colorida sublinhando as palavras. Para um deficiente visual esta representação é inacessível, mesmo que indicada na tela (que o usuário não vê). No WEBVOX o módulo de tradução processa o código HTML e indica a presença de links por um sinal sonoro. O link também recebe um número e, opcionalmente, ao final do texto é apresentada uma lista numerada com todos os links da página analisada.

A navegação pode ser realizada de quatro formas:

1. Memorizando o nome da página associada ao elo e pedindo-se explicitamente pelo nome.
2. Solicitando-se ir para o elo número N.
3. Apertando seguidamente a tecla TAB é possível obter sonoramente os nomes associados aos elos. Ao chegar ao elo desejado, aperta-se enter.
4. Durante o processo de leitura em voz, ao passar por um elo, é emitido um sinal característico. O usuário aperta Enter e nesta hora, o browser carrega a pagina referente ao último elo falado.

A navegação de retorno é feita interrompendo-se o processo de leitura e apertando uma tecla específica (backspace ou shift-TAB).

A figura 2 apresenta um trecho de código HTML com elementos comuns como títulos, links e figuras. A figura 3 apresenta a representação em um browser convencional e a figura 4 no WEBVOX.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!--saved from url=(0030)http://194.117.2.43/acessi.htm -->
<HTML><HEAD><TITLE>Acessibilidade na publicação na web</TITLE><!--
$MVD$:app("MicroVision WebExpress","769") --><!--
$MVD$:template("F:\MVAPPS\websites\mvtemp\nonio.wbt","369133e0","0") --><!--
$MVD$:color("18","808080","Gray","0") --><!MVD$:color("19","cc3333","Custom #1","0") -
-><!-- $MVD$:color("20","996666","Custom #2","0") --><!--
$MVD$:color("21","990099","Custom #3","0") --><! $MVD$:color("22","663300","Custom
#4","0") --><!-- $MVD$:color("23","80ff","Orange","0") --
-><!MVD$:color("24","ffd6ce","Lavender","0") --><!-- $MVD$:fontset("Sans
Serif","Arial","Helvetica","Monaco") --><META content="text/html; charset=iso-8859-1" http-
equiv=Content-Type> <META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD> <BODY STYLESRC="style.htm"> <H1
align=center><FONT (Continua)
```

Figura 2. Trecho de código HTML correspondente às figuras 3 e 4.

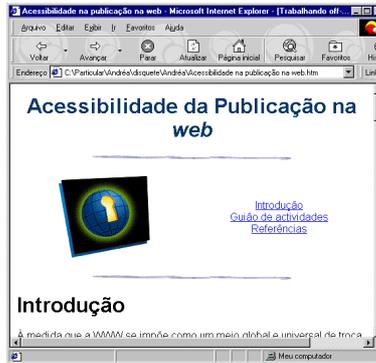


Figura 3. Imagem obtida num browser convencional.

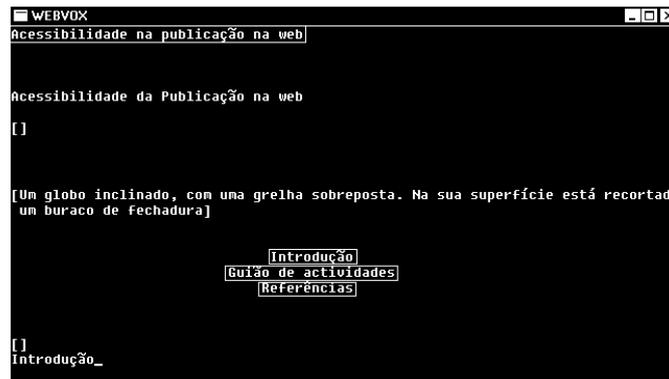


Figura 4. Aparência da interpretação no WEBVOX

8. Conclusões e Perspectivas Futuras

O Webvox foi o primeiro browser em língua portuguesa e faz parte de um projeto muito mais amplo que visa o desenvolvimento de ferramentas para a melhoria das condições de acessibilidade do deficiente visual em relação ao uso do microcomputador. Atualmente existem cerca de 3000 usuários em todo o país que utilizam o sistema Dosvox [10], e estão se beneficiando do browser implementado.

O foco inicial deste trabalho foi construir uma ferramenta que possibilitasse o acesso a Internet para os deficientes visuais através de síntese de áudio a partir de páginas HTML. Nos testes realizados com deficientes visuais notou-se que a ferramenta por si só não era suficiente. Percebeu-se então que era necessário que os desenvolvedores fossem estimulados a utilizar recursos que possibilitassem que um deficiente visual pudesse ter acesso às mesmas informações disponibilizadas para usuários normais. Isso motivou uma segunda fase do trabalho onde foi realizada uma pesquisa sobre regras de acessibilidade. Tais regras facilitam o acesso do deficiente visual as páginas Web. Em alguns países, a aplicação destas regras é obrigatória no desenvolvimento de páginas de órgãos do governo. Este tipo de obrigatoriedade

visa proporcionar aos deficientes de modo geral o acesso a Internet. Uma das contribuições deste trabalho é buscar sensibilizar os desenvolvedores no Brasil a criarem páginas acessíveis.

Uma das contribuições deste trabalho foi à realização de estudos com um grupo de deficientes visuais que permitiram avaliar a eficácia das regras propostas pelo W3C. A avaliação foi realizada com o uso de um browser descrito na seção 6. A tradução HTML-áudio realizada por este browser baseou-se nas orientações do W3C. À medida que os usuários deficientes forneciam retorno nos testes realizados, as regras de tradução foram sendo adaptadas e novas regras foram implementadas.

Os testes foram realizados com um grupo de deficientes visuais de trinta em 1999. Este grupo participou de um curso proposto com o intuito de observar e analisar o comportamento de deficientes visuais perante o computador e em relação ao uso da Internet ministrado no Centro de Tecnologia em Informática, para os alunos e professores de Instituto de Educação e Reabilitação de Cegos do Rio grande do Norte (IERC-RN.).

O protótipo implementado foi aperfeiçoado por outros membros do projeto Dosvox denominando-se hoje de Webvox II[15].

Bibliografia

1. IBM Web Accessibility for Special Needs <http://www.austin.ibm.com/sns/accessweb.html>. Novembro, 1999.
2. Microsoft Accessibility Guidelines for WWW <http://microsoft.com/enable/dev/web/guidelines.htm>. Novembro, 1999.
3. BOBBY <http://www.cast.org/bobby>. Fevereiro, 2000.
4. Web Content Accessibility Guidelines 1.0 <http://www.w3c.org/TR/WAI-WEBCONTENT>. Setembro, 1999.
5. Educação Profissional de Pessoas Portadoras de Deficiência, Relatos, Críticas e Reflexões Adonai Rocha
6. Mídia e deficiência: Manual de Estilo. Ministério da Justiça. Secretaria dos Direitos da Cidadania. Coordenadoria Nacional para Integração da Pessoa Portadora de Deficiência – CORDE. Brasília, 1996.
7. Guia de Acessibilidade. Grupo Português pelas iniciativas em acessibilidade. Internet: <http://www.acessibilidade.net>
8. WAB: World Wide Web Access for Blind and Visually Impaired Computer Users. <http://www.inf.ethz.ch>. ETH, Eidgenössische Technische Hochschule Zurich.
9. HENDRIX, Paul e BIRKMIRE, Michael. Adapting Web Browsers for Accessibility. Center for Accessible Technology. <http://www.jsrd.or.jp>
11. Projeto DosVox <http://www.nce.ufrj.br/aau/dosvox>
12. Livro acessibilidade na Internet. <http://www.acessibility.net>
13. ALMEIDA, Ana Sofia Barros. Educação Especial <http://lms.dei.uc.prática/Aulas/SF/9899/Aalmeida/interfaces.html>
14. JAMES, Frankie. Lessons from Developing Audio HTML Interfaces. Stanford University. (fjames@cs.stanford.edu)
15. PwWebSpeak for Windows 95,98 and NT. <http://www.produworks.com.pww32.htm>
16. BORGES, José Antônio, PORTO, Bernard Condorcet, SAMPAIO, Fabio Ferrentini. WEBVOX II/ INTERVOX- Um Navegador e Construtor de Páginas WEB Destinado a Deficientes Visuais. XI Simpósio Brasileiro de Informática na Educação, Maceió, 2000.

Analisando a Comunicação entre Usuários em Ambientes de Grupo

Clarissa Maria de Almeida Barbosa¹, Raquel Oliveira Prates^{2,1}, Clarisse Sieckenius de Souza¹

¹Departamento de Informática – PUC-Rio
Rio de Janeiro – RJ – Brasil

²Departamento de Ciência da Computação e Informática – IME/UERJ
Rio de Janeiro – RJ – Brasil

cbarbosa@inf.puc-rio.br, raquel@les.inf.puc-rio.br, clarisse@inf.puc-rio.br

***Abstract.** In this paper, we propose an extension to the meta-communication model [9] with the aim of enabling a more precise description of the communicative aspect of a group. We offer the designer ways of qualifying the members' utterance and specifying the structure of the discourse. We illustrate the benefits of the extension with the analysis of the communicative aspect of two different groups.*

***Resumo.** Neste artigo propomos uma extensão ao modelo abstrato de meta-comunicação do grupo [9] com o objetivo de aumentar a precisão da descrição do aspecto comunicativo de seus membros. Oferecemos ao projetista meios de qualificar a fala dos membros e de especificar a estrutura do discurso. Ilustramos os benefícios da extensão analisando a comunicação de dois grupos de acordo com o modelo original e o estendido.*

1. Introdução

O propósito de uma aplicação multi-usuário é permitir que um grupo atinja seu objetivo através da colaboração dos seus membros. Para que isto seja possível, é fundamental que eles se comuniquem para, por exemplo, executar uma tarefa em conjunto, coordenar o trabalho da equipe, permitir a socialização dos membros, etc. Assim, comunicação, colaboração e coordenação são questões centrais a serem tratadas por um sistema multi-usuário. Estas questões trazem novos desafios ao design de interfaces multi-usuário [4].

Ciente da complexidade inerente ao desenvolvimento destas interfaces, Prates (1998) apresenta um trabalho desenvolvido dentro do quadro teórico da Engenharia Semiótica [3] cuja proposta é ser um primeiro passo na direção do desenvolvimento de um ambiente que ofereça aos projetistas de interfaces multi-usuário indicadores qualitativos sobre o seu projeto.

A Engenharia Semiótica considera a interface de um sistema multi-usuário uma mensagem unilateral enviada pelo projetista aos usuários cujo conteúdo é a resposta para três perguntas fundamentais: (1) Qual é a interpretação do projetista para o problema do grupo? (2) Como seus membros devem interagir com a aplicação para resolver este problema? (3) Qual é o modelo conceitual do grupo elaborado pelo projetista? Responder essa última questão significa informar aos usuários qual é a estrutura do grupo na qual estão

inseridos, quais são as tarefas de cada membro e como elas se relacionam, assim como com quem os membros podem se comunicar e através de que linguagem e protocolos o fazem.

Através do modelo abstrato de meta-comunicação (MetaCom-G), Prates propõe o desenvolvimento de linguagens de *design* que permitam ao projetista descrever o seu modelo conceitual de grupo. A parte léxica desta linguagem é formada por unidades descritivas básicas de definição do grupo. A parte semântica, por sua vez, é composta por regras heurísticas que atuam sobre essas unidades indicando possíveis inconsistências, sem, no entanto, restringir o poder de expressão e a criatividade do projetista. Essas regras são separáveis de contexto, no sentido de que elas atuam sobre a descrição do grupo sem levar em consideração o domínio no qual ele está inserido. Portanto, o projetista é o responsável por determinar se, no domínio em questão, a inconsistência em potencial detectada é ou não uma inconsistência.

Entretanto, observamos que as unidades descritivas propostas não são suficientes para representar o complexo processo de comunicação existente em certos grupos. Neste artigo, propomos uma extensão ao MetaCom-G com o objetivo de permitir uma descrição mais precisa do aspecto comunicativo do grupo, aumentando, dessa forma, o poder de expressão deste modelo.

Na próxima seção explicaremos brevemente as unidades descritivas do MetaCom-G e descreveremos dois grupos cuja definição através dessas unidades não os caracteriza. Apresentaremos, então, a extensão proposta. Em seguida faremos uma análise de dois grupos a partir da sua descrição através do MetaCom-G estendido e mostraremos o ganho do potencial de análise obtido com a extensão deste modelo. Concluiremos o artigo apresentando as nossas contribuições e os próximos passos necessários para darmos continuidade à nossa pesquisa.

2. O Modelo MetaCom-G e a Descrição do Aspecto Comunicativo do Grupo

2.1. Dimensões de Descrição

Prates propõe que a parte léxica da linguagem de *design* seja formada por seis dimensões básicas de descrição de grupo: papéis, hierarquia, níveis de interação, objetos, capacidades de comunicação e modelos de colaboração. Como o processo de comunicação do grupo é representado apenas pela dimensão capacidades de comunicação, esta será a única a ser descrita de forma mais detalhada.

Para que um grupo funcione com eficiência e atinja seus objetivos, seus membros assumem diferentes papéis, de acordo com as suas responsabilidades. O relacionamento entre os membros e a distribuição de autoridade entre eles define a hierarquia do grupo. Os usuários de uma aplicação de grupo interagem com o sistema e entre si através da interface. Os níveis em que essa interação pode ocorrer são identificados na dimensão níveis de interação. A relação e interdependência das tarefas dos membros do grupo são capturadas na dimensão modelos de colaboração. Os objetos, por sua vez, representam tudo aquilo que faz parte da aplicação e sobre o que o usuário pode agir, ver ou falar.

As capacidades comunicativas dos membros indicam de que forma eles podem se comunicar a respeito de objetos e de outras coisas. O MetaCom-G identifica três capacidades comunicativas básicas: visão, discurso e ação [8]. Enquanto apenas os donos de um objeto são capazes de agir sobre ele, o projetista pode permitir que outros membros sejam capazes de vê-lo ou falar a seu respeito.

2.2. Necessidade de Extensão

Embora as dimensões de descrição da linguagem de *design* permitam a descrição geral do modelo do grupo, as dimensões modelo de colaboração e nível de interação são as mais exploradas. Conseqüentemente, dentre as questões centrais de uma aplicação multi-usuário, a colaboração é a mais enfatizada.

Quanto à comunicação entre os membros, é possível, através da dimensão capacidades comunicativas, descrever que membros de um grupo podem falar (um membro fala e o outro pode apenas ouvir) ou conversar (ambos os membros podem falar e ouvir) sobre determinado objeto ou tema livre. Esse poder de expressão é, no entanto, insuficiente para descrever certos grupos de forma útil e enriquecedora.

Suponha uma comunidade virtual cujo objetivo seja oferecer a seus membros a possibilidade de trocar apoio emocional e informações sobre determinada doença. O *site* drkoop.com (descrito por Preece [10]), por exemplo, hospeda várias comunidades desse tipo. Sua finalidade está relacionada à satisfação pessoal dos seus membros e é alcançada única e exclusivamente através da comunicação direta entre eles. Não existem tarefas a serem executadas. Os papéis assumidos pelos membros surgem e desaparecem durante o ciclo de vida desta comunidade, também caracterizada por não possuir uma estrutura hierárquica baseada na distribuição de autoridade entre seus membros. O relacionamento entre eles é baseado na confiança e no acúmulo de conhecimento, tenha este sido adquirido através do estudo formal ou de experiência pessoal. Todas essas características nos mostram que a comunicação entre os membros é o aspecto mais importante na descrição desta comunidade, e não a colaboração. Portanto, representar a comunicação apenas como uma conversa entre eles não é suficiente para caracterizá-la. Torna-se necessário qualificá-la, indicando, assim, que a comunidade visa oferecer apoio emocional aos seus membros. O MetaCom-G não permite essa representação.

Considere, agora, um grupo responsável por determinar a pauta de uma reunião. Em um primeiro momento, seus membros discutem sobre o assunto, sugerindo temas e expondo suas opiniões e seus comentários. Ao final da discussão, chegam a uma conclusão sobre a pauta e informam a decisão tomada às pessoas que participarão da reunião. Novamente, representar a comunicação deste grupo apenas como uma conversa entre seus membros não é suficiente para descrever o processo de tomada de decisão que caracteriza o trabalho deste grupo. O interessante, neste caso, é ser capaz de distinguir os dois processos de comunicação que compõem a definição da pauta, assim como a relação entre eles, indicando, dessa forma, que há uma discussão seguida de uma tomada de decisão.

Na próxima seção apresentaremos a extensão proposta ao MetaCom-G, que visa atender as necessidades identificadas nos exemplos acima, entre outras.

3. Propostas de Extensão do Aspecto Comunicativo do Modelo MetaCom-G

A extensão tem como objetivo permitir uma descrição mais precisa do aspecto comunicativo de um grupo, oferecendo ao projetista meios de (1) qualificar a fala dos membros e (2) especificar a estrutura do discurso no qual estarão envolvidos. Viabilizamos a qualificação da fala dos membros através da extensão das capacidades comunicativas falar e conversar atualmente disponíveis, que se mostraram insuficientes em determinadas situações. A estrutura do discurso é representada através da relação entre as capacidades comunicativas dos membros. Conseqüentemente, a descrição do grupo passa a ser feita não

apenas através das unidades descritivas básicas propostas pelo MetaCom-G, mas também através do relacionamento entre elas.

3.1. Capacidades Comunicativas (adicionando o Ato de Fala)

Propomos que a fala dos membros seja qualificada de acordo com o que está sendo feito pelo emissor ao transmitir a sua mensagem. Para isso, baseamos a nossa proposta na teoria dos atos de fala de Searle [11]. A seguir, descreveremos brevemente a nossa base teórica para, em seguida, apresentarmos a extensão proposta.

3.1.1. Teoria dos Atos de Fala

Segundo Austin [1], ao emitir uma sentença, o falante está fazendo alguma coisa, e não apenas descrevendo determinada situação. Austin identifica três atos de fala distintos produzidos pelo falante: (1) o ato locutório, ou o ato de dizer alguma coisa, (2) o ato ilocutório, produzido ao se dizer alguma coisa e, finalmente, (3) o ato perlocutório, ou o efeito causado pelo que se disse. Na sentença “Não faça isso.”, distinguimos o ato locutório “ele disse ‘Não faça isso’” do ilocutório “ele me aconselhou a não fazer isso” e do perlocutório “ele me convenceu a não fazer isso”.

Mais tarde, Searle distingue o ponto de ilocução de uma expressão da sua força de ilocução. O primeiro se refere à ação produzida ao se dizer alguma coisa, e o segundo determina a intensidade da execução desta ação. As sentenças “Você poderia fazer isso para mim, por favor?” e “Faça isso!” diferem em sua força de ilocução, mas possuem o mesmo ponto de ilocução. Searle, então, identifica cinco categorias de pontos de ilocução:

Assertivos: o falante se compromete com a veracidade da proposição, como em “Está chovendo.”;

Diretivos: o falante tem a intenção de que o ouvinte siga um curso de ação futuro. Por exemplo, “Você poderia me emprestar o seu livro, por favor?”;

Comissivos: o falante se compromete a seguir um curso de ação futuro, como em “Prometo não chegar tarde.”;

Expressivos: o falante expressa um estado psicológico sobre determinada situação. Por exemplo, “Desculpe-me pela demora em responder a sua carta.”;

Declarativos: a fala do emissor modifica a realidade do mundo, como em “Eu vos declaro marido e mulher.”

3.1.2. Qualificação da Fala

Conforme apresentamos na seção 1.2, o MetaCom-G fornece ao projetista apenas as capacidades comunicativas falar e conversar para descrever a comunicação entre os membros de um grupo. Propomos que seja oferecida, também, a possibilidade de qualificar estas capacidades de acordo com as cinco categorias de pontos de ilocução propostas por Searle: assertivo, diretivo, comissivo, expressivo e declarativo.

Essa extensão permitirá ao projetista, além de informar que um membro pode falar ou conversar sobre determinado objeto ou tema livre, indicar qual ou quais atos de fala podem ser produzidos através das suas mensagens. Tomemos o grupo de definição da pauta novamente como exemplo. Em vez de apenas representar que os membros deste grupo podem conversar sobre um tema livre, o projetista poderá dizer que eles podem realizar atos de fala assertivos, necessários para a etapa de discussão da pauta. Poderá,

também, indicar o membro responsável por expor a resolução do grupo, ou seja, quem efetuará o ato de fala declarativo¹.

De agora em diante, quando falarmos nas capacidades comunicativas dos membros, estaremos nos referindo especificamente à capacidade de discurso.

3.2. Estrutura do Discurso (adicionando o relacionamento)

A relação entre as capacidades comunicativas dos membros determina a estrutura do discurso no qual estarão envolvidos. Baseando-nos na literatura sobre relacionamento entre tarefas e ações de usuários [2][6][7], definimos quatro possíveis formas de relacionamento: seqüência, independência de ordem, intercalabilidade e concorrência.

Seqüência

As comunicações que participam deste relacionamento devem ser feitas por completo e na ordem em que aparecem. Isto significa que elas não podem ser interrompidas e devem ser feitas uma após a outra.

Suponha, por exemplo, que um professor, ciente das dificuldades enfrentadas por seus alunos, combine com cada um deles, individualmente, um horário para o esclarecimento de dúvidas. Nesse caso, todos os alunos conversarão com o professor, na ordem por ele definida, e não serão interrompidos pelos demais. Como foi marcado apenas um horário para cada aluno, aquele que já tiver conversado com o professor não poderá retornar à sua sala para esclarecer mais dúvidas, a não ser que um novo horário seja combinado.

Independência de ordem

Cada comunicação que participa deste relacionamento deve ser feita por completo antes que a próxima inicie. No entanto, diferentemente da relação seqüência, não há nenhuma restrição quanto à ordem na qual as comunicações são feitas.

Imagine, agora, que o mesmo professor, em vez de combinar um horário fixo com cada aluno, tenha optado por ficar uma tarde inteira à disposição deles. Agora cabe aos alunos se organizarem ou não para conversar com ele. O professor, no entanto, garante que enquanto um aluno estiver falando com ele, este não será interrompido pelos demais, e que todos conseguirão esclarecer suas dúvidas naquela tarde.

Intercalabilidade

Em uma relação intercalabilidade, as comunicações podem ser interrompidas pelas demais e retomadas posteriormente.

A relação entre professor e alunos durante uma aula é um bom exemplo. Ao ensinar a matéria, o professor pode ser interrompido pelos alunos, caso estes tenham dúvidas ou desejem fazer algum comentário. Após o esclarecimento da dúvida, o professor continua a matéria do ponto onde parou.

¹ Consideramos a decisão sobre a pauta um ato de fala declarativo, uma vez que a pauta só passa a existir após esta tomada de decisão.

Concorrência

As comunicações envolvidas nesta relação podem ser feitas simultaneamente. Imagine um grupo de amigos comemorando o aniversário de um deles em um restaurante. Tipicamente, nesta situação, ocorrem conversas paralelas. No entanto, em um ambiente formal, como em uma reunião de negócios, por exemplo, deve-se ter cuidado ao permitir conversas paralelas, pois representam um risco para o alcance do objetivo do grupo.

Vale a pena ressaltar que as relações intercalabilidade e concorrência indicam que existe a **possibilidade** de haver interrupção e simultaneidade entre as comunicações.

Toda relação entre comunicações define um discurso. Esse discurso, por sua vez, pode ser combinado com outra comunicação ou discurso através de um relacionamento, formando um novo discurso, e assim por diante. Esses agrupamentos permitem que um processo de comunicação complexo seja definido em um nível de abstração maior. A abstração facilita a compreensão do processo de comunicação e torna a sua análise mais eficiente, pois permite-nos pensar nele como um todo, sem nos envolvermos com seus detalhes.

É importante observar o impacto da representação da estrutura do discurso na linguagem de *design* proposta pelo MetaCom-G. Atualmente, essa linguagem é composta por uma parte léxica, formada pelas unidades descritivas básicas de definição de grupo, e semântica, composta pelas regras heurísticas. Como a estrutura do discurso é representada através da definição das combinações válidas das capacidades comunicativas dos membros, um elemento da parte léxica da linguagem, a sua representação implica a definição de um novo nível de descrição do grupo, o sintático. A linguagem de design passa a ser, então, formada por uma parte léxica, uma sintática e uma semântica.

3.3. Sintaxe (nova representação)

As extensões propostas são representadas da seguinte forma:

Capacidades comunicativas

fala(<falante>, <assunto>, <ouvinte>, <ato_de_fala>)

conversa(<lista_membros>, <assunto>, <ato_de_fala>)

Estrutura do discurso

discurso(<nome_discurso>, <relacionamento>, <lista_componentes_discurso>),

onde:

<componentes_discurso>={<capacidades_comunicativas>|<nome_discursos>}

* * *

Retomemos o exemplo do grupo de definição da pauta de uma reunião. Dissemos que o processo de tomada de decisão que caracteriza o seu trabalho é composto por uma discussão seguida da tomada da decisão. Imagine que este grupo seja formado por três pessoas. A representação das capacidades comunicativas dos membros e da estrutura do discurso da tomada de decisão é apresentada abaixo:

Membros

m1, m2, m3

Capacidades comunicativas

conversa((m1,m2,m3),pauta,assertivo)

fala(m1,pauta,grupo_afetado,declarativo)

Estrutura do discurso

discurso(discussao,intercalabilidade,conversa((m1,m2,m3),pauta,assertivo))

discurso(processo_tomada_decisao,seqüencia,(discussao,fala(m1,pauta,grupo_afetado,declarativo)))

O processo de tomada de decisão é descrito em um nível de abstração maior, já que não apresenta os detalhes da etapa de discussão. Essa descrição permite-nos pensar neste processo como um todo e, conseqüentemente, analisá-lo sem conhecermos os seus detalhes. Caso seja necessário, pode-se, posteriormente, analisar a etapa de discussão da pauta isoladamente.

4. Análises com as Extensões do MetaCom-G em Comparação com o Modelo Original

A descrição de um grupo pode ser utilizada como fonte de informação para uma análise do mesmo. Os aspectos que podem ser examinados e a profundidade da análise dependem das informações sobre o grupo disponíveis na sua descrição. Nesta seção faremos a análise de três grupos a partir da sua descrição através do MetaCom-G estendido e mostraremos o ganho do potencial de análise obtido com a extensão.

Como estamos interessados no aspecto comunicativo do grupo, as descrições aqui apresentadas são feitas apenas em função das capacidades comunicativas e da estrutura do discurso. A adoção de termos bastante genéricos é proposital, pois desejamos deixar claro que a análise do grupo é feita a partir da sua descrição, independente do domínio da aplicação. Na prática, no entanto, sugerimos a utilização de termos do domínio para representar as estruturas necessárias.

4.1. Caso 1: Conselho Consultivo x Conselho Deliberativo

Considere a descrição de dois grupos, A e B, conforme apresentada abaixo:

Grupo A

Membros

m1, m2, m3

Capacidades comunicativas

conversa((m1,m2,m3),assunto,assertivo)

fala(m1,assunto,ouvinte,assertivo)

Estrutura do discurso

```
discurso(discurso1,intercalabilidade,conversa((m1,m2,m3),assunto,assertivo))
```

```
discurso(discurso2,seqüência,(discurso1,fala(m1,assunto,ouvinte,assertivo))
```

Grupo BMembros

m1, m2, m3

Capacidades comunicativas

```
conversa((m1,m2,m3),assunto,assertivo)
```

```
fala(m1,assunto,ouvinte,declarativo)
```

Estrutura do discurso

```
discurso(discurso1,intercalabilidade,conversa((m1,m2,m3),assunto,assertivo))
```

```
discurso(discurso2,seqüência,(discurso1,fala(m1,assunto,ouvinte,declarativo))
```

A estrutura de `discurso1` dos grupos A e B é a mesma. Analisando-a, observamos que este discurso é formado por conversas entre todos os membros e sabemos, pela relação intercalabilidade, que essas conversas se alternam. O ato de fala assertivo dessas conversas nos indica que os participantes estão contando aos demais como as coisas são [11], por exemplo, reportando fatos e situações, expondo suas opiniões, etc. Essa estrutura é típica de uma discussão entre um grupo de pessoas a respeito de determinado assunto.

A relação seqüência que define a estrutura do `discurso2` de ambos os grupos nos informa que ele é composto por uma discussão entre os membros, seguida da fala de um dos membros que participou da discussão. O fato de que um dos componentes de `discurso2` é, por sua vez, um discurso, permite-nos saber que `discurso2` é um processo de comunicação mais complexo do que `discurso1`. A diferença entre os dois grupos está no ato de fala expresso pela fala de m1. Enquanto o de A é assertivo, o de B é declarativo, indicando que este, ao transmitir a sua mensagem, provoca alterações no estado do mundo.

Analisando `discurso2` de A, observamos que, após a discussão, um dos membros reporta a um ouvinte o que foi anteriormente discutido. Essa estrutura caracteriza o trabalho de um Conselho Consultivo, um grupo acionado toda vez que se deseja conhecer a sua opinião sobre determinado assunto. Seus membros se reúnem para discuti-lo, chegam a uma conclusão e repassam-na à pessoa ou ao grupo que solicitou a informação.

O `discurso2` de B, por sua vez, diz que, após a discussão, m1 muda o estado do mundo baseado no que foi discutido anteriormente. Dessa forma, sabemos que enquanto o grupo A, durante a reunião, chega a uma opinião sobre o assunto em questão, B toma uma decisão. A estrutura de `discurso2` de B caracteriza o processo de tomada de decisão de um Conselho Deliberativo. Seus membros conversam sobre o assunto, tomam uma decisão e comunicam-na à pessoa ou ao grupo interessado.

Vamos, agora, comparar a análise acima com a obtida a partir da descrição dos grupos A e B utilizando o MetaCom-G. A definição do aspecto comunicativos desses grupos através deste modelo é a mesma e está representada abaixo:

Membros

m1, m2, m3

Capacidades comunicativas

conversa((m1,m2,m3), assunto)

Essa descrição permite-nos conhecer apenas as capacidades comunicativas dos membros de cada grupo. No entanto, não temos informações sobre os processos de comunicação que ocorrem em ambos os grupos, nem o que está sendo feito através desses processos. Conseqüentemente, não conseguimos distinguir os grupos A e B a partir da sua descrição pelo MetaCom-G.

Quando descrevemos o grupo de definição de pauta (seção 1.2), dissemos que seria interessante distinguir os dois processos de comunicação que compõem o processo de definição da pauta: a discussão e a tomada de decisão. Observe que o processo de tomada de decisão que caracteriza o trabalho deste grupo é o mesmo do que descreve o trabalho do Conselho Deliberativo, que conseguimos representar através do MetaCom-G estendido. Dessa forma, as extensões propostas também atendem à necessidade surgida ao analisarmos o grupo de definição da pauta.

4.2. Caso 2: Comunicação Expressiva Intercalada

Considere, agora, a seguinte definição de um grupo:

Grupo X

Membros

m1, m2, m3, m4

Capacidades comunicativas

conversa((m1,m2,m3,m4), assunto, expressivo)

Estrutura do discurso

discurso(discurso1, intercalabilidade, conversa((m1,m2,m3,m4), assunto, expressivo))

O ato de fala expressivo que caracteriza o discurso deste grupo nos informa que seus membros podem expressar seu estado psicológico em relação a determinada situação. Expressar solidariedade, compaixão, compreensão, etc. é um comportamento típico de grupos cujo objetivo é a troca de apoio emocional entre seus membros. Através da análise da comunicação disponibilizada aos membros de comunidades virtuais, podemos, por exemplo, identificar aquelas que possuem essa finalidade, como era o caso da comunidade de saúde apresentada na seção 1.2.

Para mostrarmos o aumento do potencial de análise deste grupo, obtido com a extensão do MetaCom-G, compararemos a análise acima com a resultante da descrição do grupo X utilizando o modelo original.

Membros

m1, m2, m3, m4

Capacidades comunicativas

conversa((m1,m2,m3,m4), assunto)

Como a descrição do grupo através do MetaCom-G permite-nos apenas conhecer as capacidades comunicativas de seus membros, nada podemos inferir sobre os processos de comunicação que nele ocorrem.

5. Conclusões e Trabalhos Futuros

Neste artigo estendemos o MetaCom-G proposto por Prates, com o objetivo de permitir uma descrição mais precisa do aspecto comunicativo de um grupo. Qualificamos a dimensão capacidades comunicativas para oferecermos ao projetista meios de especificar os atos de fala que podem ser expressos pelos membros. Além disso, definimos um novo nível de descrição na linguagem de *design*, o sintático, que permite ao projetista especificar a estrutura do discurso no qual os membros estarão envolvidos. O MetaCom-G estendido, portanto, aumenta o poder de expressão do projetista ao comunicar o seu modelo conceitual do grupo.

O ganho imediatamente obtido com a extensão do MetaCom-G é o aumento do potencial de análise do grupo, em tempo de *design*. Através da análise do aspecto comunicativo de um grupo, podemos identificar as estruturas de comunicação disponíveis aos seus membros e, com isso, caracterizar aspectos da interação entre eles. Esta análise fornece, também, outros indícios sobre o grupo, como, por exemplo, seu objetivo, regime político, modelo de colaboração, etc. Essas informações são úteis para o projetista, pois além de permitirem que ele verifique se o grupo descrito é realmente o desejado, elas o ajudam em futuras etapas do processo de *design*. De posse de maiores informações sobre os processos de comunicação do grupo, o projetista pode, por exemplo, definir a ferramenta de comunicação (*chat*, lista de discussão, *e-mail*, entre outras) apropriada para dar suporte a estes processos e os *widgets* mais adequados ao seu funcionamento [10], aumentando, dessa forma, as chances de que o grupo atinja seu objetivo de forma eficiente.

Os exemplos de análise descritos neste trabalho mostram o aumento da capacidade de expressão possibilitado pela extensão apresentada. No entanto, ainda é necessário fazer uma avaliação mais extensa da nossa proposta. Devemos investigar se os relacionamentos oferecidos são suficientes para representar as estruturas de conversação desejadas. Precisamos, também, verificar se seria interessante oferecermos ao projetista a possibilidade de qualificar as falas dos membros em um nível de abstração inferior ao alcançado com a qualificação pelos atos de fala. Durante a etapa de discussão da pauta, por exemplo, os participantes submetem propostas, comentam-nas e votam nas que julgam ser mais urgentes. Através do MetaCom-G estendido não é possível distinguir essas comunicações, pois as três expressam o mesmo ato de fala, o assertivo. No entanto, ao analisarmos as vantagens de uma qualificação mais detalhada da fala, devemos ter em mente que o MetaCom-G propõe que as unidades descritivas básicas de definição de grupo sejam

definidas independentemente do domínio. Devemos, portanto, verificar se qualificar a fala em um nível de abstração inferior ao proposto nesse artigo não implica em romper com esse compromisso.

A proposta de Prates é que se utilize o MetaCom-G para dar apoio ao *design* de interfaces multi-usuário. Para que o MetaCom-G estendido também possa ser usado dentro da arquitetura de *design* por ela proposta, é necessário definir as novas regras semânticas, também separáveis de contexto e descritivas, que fornecerão ao projetista indicadores qualitativos sobre a comunicação do grupo. Por exemplo, as regras “membros que podem expressar atos de fala comissivos devem, também, poder expressar atos de fala assertivos” e “membros que sofrem atos de fala diretivos devem poder expressar atos de fala assertivos” alertam o projetista sobre a necessidade de se dar um retorno aos membros com quem determinado membro se comprometeu.

As análises apresentadas neste trabalho caracterizam a descrição das possíveis comunicação do grupo em tempo de *design*. Cabe ressaltar que a análise de um grupo durante a sua vida (ou seja, em tempo de execução) é também muito interessante. Pode-se, por exemplo, a partir da estrutura retórica do discurso, identificar os atos de fala que estão efetivamente sendo expressos pelos membros e, assim, traçar o perfil do grupo. Essa informação poder ser útil para um usuário decidir se quer ou não tornar-se um membro. O coordenador do grupo, por sua vez, pode utilizá-la tanto para verificar se o grupo está interagindo de acordo com o seu objetivo, como para planejar a sua evolução [10].

6. Referências

- [1] Austin, J.L., How to Do Things with Words, Cambridge: Harvard University Press, 1962.
- [2] Bolognesi, T., Brinskma, H., Introduction to the ISO Specification Language LOTOS, Computer Networks and ISDN Systems, Vol.14, pp.25-59, 1987.
- [3] de Souza, C.S., The Semiotic Engineering of User Interface Languages, International Journal of Man-Machine Studies, Vol.39, pp.753-773, 1993.
- [4] Ellis, C., Gibbs, S. and Rein, G., Groupware: Some Issues and Experiences, CACM, Vol.34, pp.39-58, 1991.
- [5] Grudin, J., Groupware and Social Dynamics: Eight Challenges for Developer, CACM, Vol.37, pp.93-58, 1994.
- [6] Hix, D. and Hartson, H.R., Developing User Interfaces: Ensuring Usability Through Product and Process, New York: John Wiley & Sons, 1993.
- [7] Paternó, F., Model-Based Design and Evaluation of Interactive Applications, Chapter4, pp.39-53, Springer, 1999.
- [8] Prates, R.O., de Souza, C.S., and Garcia, A.C. B., A Semiotic Framework for Multi-User Interfaces, SIGCHI Bulletin, Vol.29, pp.28-39, 1997.
- [9] Prates, R.O., A Engenharia Semiótica de Linguagens de Interfaces Multi-Usuário, Tese de Doutorado, Departamento de Informática, PUC-Rio, 1998.
- [10] Preece, J., Online Communities: Designing Usability, Supporting Sociability, New York: John Wiley & Sons, 2000.
- [11] Searle, J., Expression and Meaning: Studies in the Theory of Speech Acts, Cambridge: Cambridge University, 1979.

Analyzing HCI in Multiple Environments: A Case Study

Simone Diniz Junqueira Barbosa, Clarisse Sieckenius de Souza, Gilda Dahis

Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
Gávea, Rio de Janeiro, RJ 22453-900
{sim, clarisse, gilda}@inf.puc-rio.br

Abstract. *As the computer industry moves towards universal accessibility, with a wide variety of interactive devices and environments, the need for specific research in HCI analysis and design for multiple environments grows accordingly. In this paper, we investigate some communicative aspects of interactive applications, borrowing concepts from communication-related disciplines. Our goal is to take the first steps in coupling lessons learned in these disciplines with a theoretical study of HCI, in order to devise a framework of analysis that can be applied to diverse environments.*

Keywords: *multiple-environment interfaces, interaction model, discourse structure, pragmatics, HCI analysis, semiotic engineering*

1. Introduction

Over the last few years, we have been watching the computer industry develop an ever-increasing variety of interactive environments and devices, in an attempt to reach the largest possible number of users, and thus move towards universal accessibility. The wide range of input and output capabilities brings about an urgent need for specific research in HCI analysis and design for multiple environments. On the one hand, we have environments with diverse characteristics, each demanding a particular HCI treatment. On the other hand, we often need to provide the same services and systems across environments, which requires a more integrated approach to multiple-environment HCI analysis and design.

Most interaction models used in HCI focus on the grammatical aspects of well-behaved interaction. Some researchers have also provided guidelines that aim at achieving successful interaction. However, most of them have not taken advantage of the communicative aspect of interactive software, and have not looked into linguistic or semiotic literature for investigating the similarities between HCI and these fields. For instance, Linguistics has long investigated the complementary roles of grammar and pragmatics in language, as we can see in the following excerpt:

“Language consists of grammar and pragmatics. Grammar is an abstract formal system for producing and interpreting messages. General pragmatics is a set of strategies and principles for achieving success in communication by the use of grammar.” (Leech, 1983:76)

Viewing interaction from a communicative perspective, we could try to use studies from other communication-related fields to provide a theory of HCI, which is mostly an applied field of research. One such theory has come forth under the name of Semiotic Engineering

(de Souza, 1993). It views a computer system's interface as a unique one-shot meta-message from designer(s) to users. It is one-shot insofar as the designer is no longer available for carrying out a genuine two-way conversation with users about the conveyed message. It is a meta-message, because it is itself capable of generating and exchanging with users a number of messages about the interaction, and which make up the interaction proper. It is also unique, because each computer application is built to solve particular problems of a given domain, following a specific design strategy, which cannot be found in its entirety in any other application.

By investigating the characteristics of interaction in different environments, one of our goals is to analyze the advantages and disadvantages of having a single framework for analyzing (and designing) interaction of a common range of services in different media and environments. In order to guide our reflection about these issues, we present a case study, through which we have reached some interesting conclusions and further questions.

In this paper, we apply linguistic studies of discourse structures and pragmatics to an analysis of HCI phenomena in multiple environments. We do so while adopting the semiotic engineering theoretical framework as the basis for our work.

In previous work (Dahis, 2001), we have developed a model for representing interaction in such a way as to abstract details of specific interface styles and technological constraints, following an **interaction-as-conversation** paradigm (Brennan, 1990). This model, however, did not explore some relevant aspects of discourse structure nor pragmatics. In this paper, we investigate how this model, coupled with some studies about discourse structure (Gross & Sidner, 1986; Wachtel, 1986) and pragmatics (Leech, 1983), could be used to analyze interaction for a single set of services in different environments.

2. Interaction Model

Dahis (2001) has proposed an interaction model for representing interaction-as-conversation. Her goal was to provide a common model for representing interaction at a high level of abstraction, which could be instantiated into different environments and interface styles. According to her, having a common interaction model would facilitate the design and maintenance of these interfaces.

Dahis' model was created to provide a means for modeling interaction scenarios in a semi-formal representation. Her model represents interaction at three levels of abstraction. From higher to lower levels, we have: tasks, dialogs, and utterances.

A task in this model represents how a real domain task may be accomplished using the system, *i.e.* a complete interaction scenario. It encompasses one or more dialogs (stages of interaction), which in turn are composed of utterances (units of interaction). She has classified all elements into semantic types at each level, and created possible mappings between levels. In order to do that, she devised one or more templates of dialogs and corresponding utterances for any given dialog type. These templates were written in an interaction scenario specification language she has created, called LECI.

A typical task specification in LECI would be:

```

Task CheckBalanceAccount (Domain Information)
sequence
{
  Dialog AccountNumberPassword (Information Request)
  Dialog ServicesSelection (Selection)
  Dialog AccountServices (Selection) <includes ServicesSelection>
  Dialog AccountBalance (Domain Information) <includes
ServicesSelection>
}

```

where CheckBalanceAccount is the task name, Domain Information is the semantic type of the task. This task is composed of the dialogs AccountNumberPassword, ServicesSelection, AccountServices, and AccountBalance, each one with its corresponding semantic type. The <includes dialog> clause indicates that dialog is made available simultaneously within the current dialog. The term **sequence** indicates that the dialogs between system and user will be exchanged in the specified order.

Dahis' language has a grammar in which designers may specify the syntax and semantics of interaction conversations. However, until now it has not addressed in depth some important pragmatic issues that may guide designers as to how to combine these components into a **usable** dialog structure. A comprehensive study of this matter was left for future work on her dissertation.

In addition, although she claimed to be media- and environment-independent, she provided little evidence to support this claim. In this paper, we will present a case study in which we analyze an application in three different environments, and raise a few questions about how her model could be enriched by analyzing linguistic aspects, in particular through discourse analysis and pragmatics.

3. Discourse Analysis & Pragmatics

Discourse analysis can be said to study "how humans use language to communicate and, in particular, how addressers construct linguistic messages for addressees and how addressees work on linguistic messages in order to interpret them." (Brown & Yule, 1983:ix). Several views can be considered, viewing sentence-as-object, text-as-product, and discourse-as-process. As one may guess, the latter view is much in line with Semiotic Engineering. In it, sentences in a discourse are considered to be an evidence of an attempt by a speaker/writer (message producer) to communicate a message to a hearer/reader (message recipient). Both production and interpretation processes are studied as two ends of a larger communication process.

Some of the issues we find interesting to investigate in HCI by making a parallel with studies in related communication fields are: topic of conversation, topic-shift, turn taking, references and anaphors, context, co-text, presuppositions, implicatures, and so on. We would like to refer the reader to Brown & Yule for an in-depth discussion about these concepts. In this paper, we will approach some of these issues as they are brought up in our case study.

In considering interaction from a conversational perspective, we may also find it useful to try to apply studies from Pragmatics to HCI. In particular, Grice (1975) has

proposed that a conversation should follow a general principle called the *Cooperative Principle* (CP):

“Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.”
(Grice, 1975:45)

This principle is supported by a number of conversational conventions, or *maxims*:

Maxim of Quantity: Make your contribution as informative as is required (for the current purposes of the exchange). Do not make your contribution more informative than is required.

Maxim of Quality: Do not say what you believe to be false. Do not say that for which you lack adequate evidence.

Maxim of Relation: Be relevant.

Maxim of Manner: Be perspicuous. Avoid obscurity of expression. Avoid ambiguity. Be brief (avoid unnecessary prolixity). Be orderly.

In addition, Grice proposes that a maxim such as “Be polite” is also normally observed.

In the next section, we present a case study in which we analyze multiple environments for a single set of services, in view of the aforementioned interaction model and pragmatic principles.

4. Banking Environments: A Case Study

In our case study, we chose the application of a banking system, due to its immediate availability in multiple environments: phone banking, ATM, and web banking. The task was to check the balance of one of his checking accounts. During the test, we took notes about how the user carried out the interaction, specified it in the LECI language, and made some brief discourse analysis of the conversation that was exchanged between user and system. Our goal was twofold: first, we wanted to check if LECI could be used to specify an application's interaction possibilities at a high level of abstraction (independently of the environment), that would later be specialized or instantiated according to the environment. Second, we wanted to investigate if a discourse analysis could complement the analysis based on the LECI language.

4.1 Phone System

Figure 1 illustrates the interaction that took place in the phone-banking system, and the corresponding LECI specification (to the right):

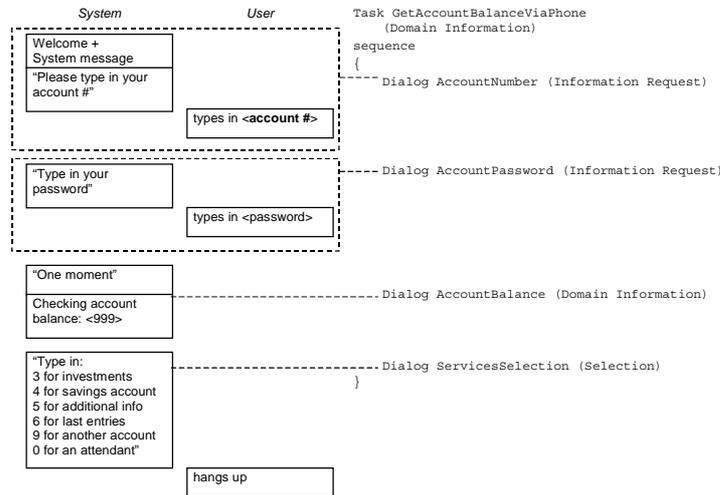


Figure 1: Phone interaction for getting a checking account balance, and the corresponding interaction specification.

One may notice that the user provides the account number, and the system does not utter it back in order to confirm it. Instead, the information that the account number is correct is implicit because of the password validation. This might seem a violation of the Maxim of Quantity. However, since the rate of speech production is fast, the topic of the conversation, which is the account number, is retained in memory as it would in normal human conversation. In spite of this, if a user has more than one checking account at one time, he/she may forget which account is being verified at the moment, and there is no means for him/her to retrieve the current topic of conversation in this system; he/she would have to start over, and pay more attention next time.

4.2 ATM System

Figure 2 illustrates the interaction that took place in the ATM system for the same task (getting the checking account balance), and the corresponding LECI specification (to the right). This interaction takes much longer than the previous one, with the phone-banking system. In part, this is due to security issues (asking for the password before sensitive information is provided), but also because there are more services available at the ATM than by the phone-banking system, i.e. the two systems are not functionally equivalent. One may notice how the service is chosen, by a composition of three consecutive choices, as expressed by the *ServicesSelection* dialog.

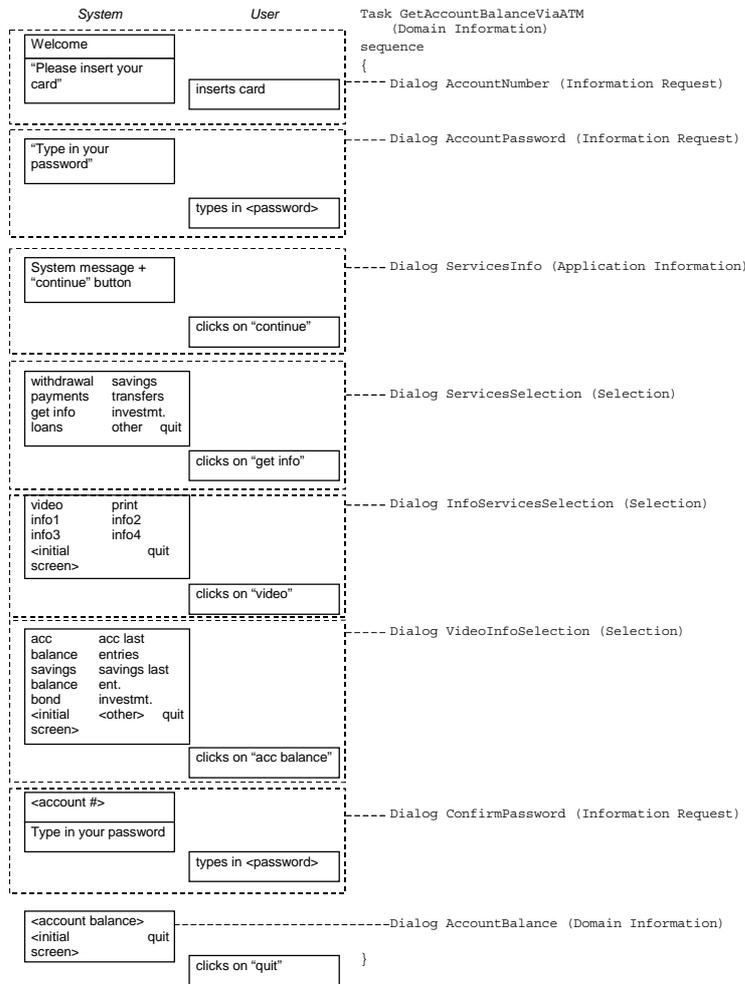


Figure 2: ATM interaction for getting a checking account balance, and the corresponding interaction specification.

In the ATM, the card the user inserts in the ATM slot provides the account information. While choosing the desired service, the user has no information about which account he/she is dealing with. But, when it comes the time to type in a password for confirming the chosen operation, the system provides the account number, so that the user may explicitly recall this information. This explicit recall is adequate (in agreement with the Maxims of Quantity and of Relation), since the ATM requires that the user navigate through a series of screens before getting the information he/she needs, and he/she might as well forget the account he/she is dealing with. However, when providing the account balance, this information is no longer available. So, if the user has the same password for

all his/her accounts, and types in the password without looking at the account number, he/she is no longer able to retrieve it when the desired information is provided. This problem reflects a violation of Grice's Maxim of Quantity, because not all required information is presented at one conversational exchange.

Another issue we may notice in this example is the early selection of displayed/printed information. After the user reaches some information displayed on video, the only way for him/her to get the same information printed on paper is to start over, and follow the alternative "print" path.

4.3 Web System

Although the web-banking system offers the widest range of functionalities among the investigated systems, we observe that interaction is brief. This is possible because of the openness of web pages. We no longer have limitations due to the linearity of the channel or small display space. This makes it possible not only to offer a wide range of options at once, but also to explicitly provide the topic of the conversation, *i.e.* the account number, at each interaction step. If the user needs to recall it, no effort is required but that of perceiving a small piece of information that is consistently presented at the same location on the pages throughout the system.

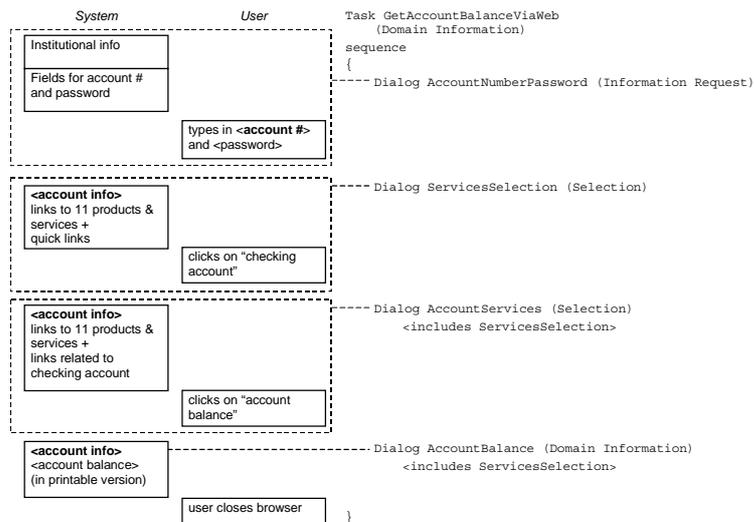


Figure 3: Web interaction for getting a checking account balance, and the corresponding interaction specification.

We have also observed a large amount of institutional information and advertisement of additional products and services on the web system. Considering most users' purposes of checking an account balance or accomplish some other transaction, this additional information constitutes a violation of the Maxim of Relation.

One characteristic of web applications is the ability to open more than one window, *i.e.* to carry out simultaneous conversations. Whereas, in natural conversation, we are constrained to a single thread of conversation at any given moment (with the possibility of

switching back and forth between topics), we are now completely unconstrained (except due to technological limitations such as memory capacity, which we will not consider here). This can be a major source of confusion and frustration to inexperienced users, for it does not have any parallel in human-to-human conversation.

4.4 Interaction Specifications

In order to compare the interaction in the three different environments, as specified in Dahis' specification language, LECl, we reproduce below the specifications, where the semantic types (shown previously between parentheses) have been omitted for the sake of clarity:

Phone	ATM	Web
<pre> Task GetAccountBalanceViaPhone sequence { Dialog AccountNumber Dialog AccountPassword Dialog AccountBalance Dialog ServicesSelection } </pre>	<pre> Task GetAccountBalanceViaATM sequence { Dialog AccountNumber Dialog AccountPassword Dialog ServicesInfo Dialog ServicesSelection Dialog InfoServicesSelection Dialog VideoInfoSelection Dialog ConfirmPassword Dialog AccountBalance } </pre>	<pre> Task GetAccountBalanceViaWeb sequence { Dialog AccountNumberPassword Dialog ServicesSelection Dialog AccountServices <includes ServicesSelection> Dialog AccountBalance <includes ServicesSelection> } </pre>

From the interaction specifications, we may notice that, on the phone, the *default* operation is to verify the account balance, and thus no user selection must be made. This is probably due to the limitations of the linearity of the channel, coupled with cost considerations. Moreover, all the services in this system are for information retrieval only; no transactions can be made without a human attendant. In contrast, on the ATM and on the web, a whole range of services is offered. A couple of things are worthy of comment: how user identification was handled, how on-screen and printable versions of the information were accessed, and how the navigation between selection options could be made.

User identification took place differently in each environment. In phone banking and web banking, the user provided the account number, whereas in the ATM, the card itself provided the identification. In both systems the user had to confirm his/her id by means of a password. On the phone system, the account number and the password were requested in two consecutive moments, whereas, on the web, they were provided at the same interaction exchange. As soon as the identification took place, the range of possible conversations was constrained by both user profile and technological considerations.

The forms of access to on-screen and printable versions of information also differed between systems. On the web, the information was displayed on-screen, and the user could print it through extra-system features of the browser. On the ATM, the user had to prematurely choose the media in which the information should be provided. This difference reveals an absence of an underlying high-level interaction model common to both systems.

As to the navigation structures, which somewhat reflect the dialog structures, we find a narrow and shallow structure on the phone system, *i.e.* few options are presented at once, and there are few options overall. On the ATM, we have many services options, but due to display size limitations, few are presented at any given moment. In other words, the ATM presents a narrow but deep structure. The web system, virtually unconstrained, presents a wide and shallow structure, in addition to a persistent menu of services, which provide direct access to any second-level menu in the system.

5. Concluding Remarks

In the environments under investigation, we have observed great variations on the length of conversations (reflected on the access structures), amount of information in each conversational exchange, and range of topics that can be explored (amount of services offered). For instance, our phone-banking system provided only a small subset of the functionalities and a small subset of the information items present in the ATM and web-banking system. We have found that, moving from the phone-banking system, to the ATM, and then to the web-banking system, discourse goes from terse to verbose. These differences lead to a need for diverse dialog structures to support each environment's characteristics.

The amount of information presented at each exchange in both the phone-system and the ATM was equivalent. As a result, the phone system could have provided as much functionality as the ATM, except maybe due to security restrictions. Figure 4 illustrates the amount of information provided in each environment, considering the actual and potential amount (where *potential* refers not to the physically possible, but to what we find pragmatically adequate to each environment).

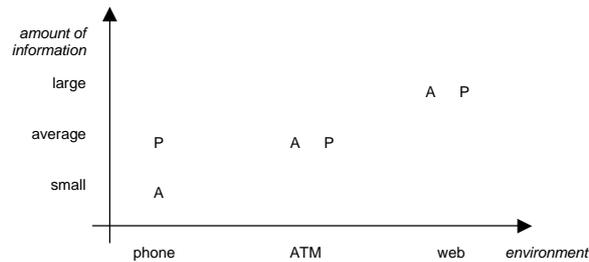


Figure 4: Amount of information potentially (P) and actually (A) provided in each environment.

The amount of information provided on the phone “at each exchange” is distributed in a short period time, due to the linearity of the channel. In contrast, on the ATM and on the web, various chunks of information may indeed be presented at once. Due to display space limitations on the ATM, sometimes one or two navigation steps are required. On the web, the space is virtually unconstrained. In this case, human cognitive limitations provide the constraints, for it may be strenuous for us to be presented with every single available alternative at once, and have to look for the desired operation or content in a large, dense page.

Due to the high rate of speech production and linearity of the phone system, the user had to maintain the topic of conversation in mind, making use of his/her recent memory. This suggests that the length of conversation must be short, at risk of having the user forget what he/she is talking about. In the ATM, it is the limited display space that does not allow much contextual information, and users also have to make use of their recent memory for contextualizing some of the operations. In lengthy exchanges, the topic must be presented at strategic points in the conversation, in order to facilitate its recall in intermediate exchanges or interaction steps. In contrast, in web banking, the system is

always able to provide contextual information about the current topic of discourse. The persistence of information requires no effort of recall on the part of users.

We have noticed that conversation on the web is less focused. It is up to the user to select and restrict the focus of conversation from a wide range of possibilities. On the phone, however, the sequential nature of the channel has to been taken into account, and a more focused discourse took place.

Moreover, the web makes it possible to carry out multiple conversations in parallel, which is not possible on the other systems. Natural discourse itself has a sequential nature and does not allow for such parallelism. Instead, it provides means for switching back and forth between conversation themes and topics. On the web, however, this is not necessarily so. The user may open several different windows and carry out a number of independent conversations almost simultaneously, which may cause confusion and frustration to users. We may then consider that the phone system is linear, whereas the ATM is bi-dimensional, and the web system is n-dimensional (Figure 5). In order to avoid the undesirable side effects of the n-dimensionality of web technology, developers may explicitly constrain the application.

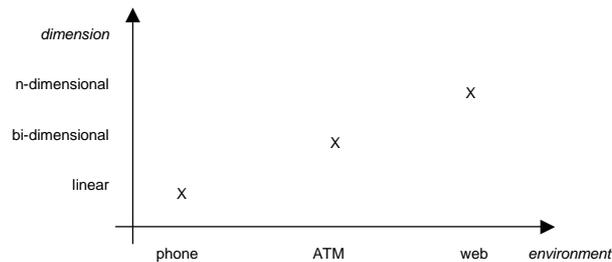


Figure 5: Conversation dimensions in different environments.

When it comes to references, users of a phone-banking system have not only to keep in mind the theme and topic of conversation, but also explicitly state that they want to “go back” to a previous topic.

In natural conversation, the whole range of recently referenced topics can be accessed at any time. In computational systems, however, users need access mechanisms for this. Due to the sequential nature of conversation in phone-banking systems, users may only traverse these access structures one step in each direction: 1) choose a topic of conversation or further constrain a topic; or 2) go back to a higher level in the structure, to a previous topic. Moreover, sometimes it takes quite a few exchanges in order to complete the selection of the topic, i.e., the sentence generation follows a compositional approach. In this case, the access structure may be too deep, and users may be caught in intermediary steps, with a yet-undefined topic, and no present indication of what he/she has accomplished up to that point. The ATM follows this access structure closely, but the options available at one time are usually all displayed simultaneously on a “page”. (Sometimes users must navigate through a couple of “pages” due to a large number of options). Depending on the amount of display space, the number of options presented at a time can be significantly large, and thus we may avoid too deep access structures. On the web, on the other hand, display space does not pose a considerable

limitation, and almost all of the available options are presented at once, providing an access structure that is shallow and wide shallow (maybe too wide).

In natural conversation, people can always switch from talking about a topic to talking about the conversation itself, and then back to the topic. These *inserted* sequences may be used to reinforce or recall the current topic. None of the investigated systems allowed for inserted sequences, which would be represented by a help or explanation system.

Table 1 summarizes some of the characteristics we found in analyzing the banking system in different environments.

	Natural	Phone	ATM	Web
dialogue style	verbose	terse	terse	verbose
dialogue structure	natural	narrow & deep; compositional	balanced	shallow & wide
sequencing	single-track, linear	single-track, linear	single-track, bi-dimensional	multi-track, multidimensional
treatment of anaphors and references	"object" may be either implicit or explicit; may be memorized or explicitly recalled	"object" is implicit; must be memorized	"object" is explicit and has a physical manifestation	"object" is explicit
turn-taking & identification of current speaker	natural exchanges	"full-duplex" (system may be interrupted at any time)	well-marked (system displays info; user makes a selection, types in data or inserts card)	well-marked (system displays info; user makes a selection or types in data)
time/space constraints	temporal & cognitive constraints	temporal & cognitive constraints	physical, 2-D spatial constraints	almost unconstrained
end-of-conversation signaling		user hangs up	explicit <Quit> button	user closes browser window (extra-application) or clicks on explicit <Quit> link
initiative	mixed	mixed (if idle for long, the system automatically calls an attendant or hangs up)	fixed	fixed

Table 1: Characteristics of conversation found in our case study.

In this paper, we have seen that a high-level interaction model can be complemented with communication-related studies for enriching the analysis of HCI in multiple environments. An interaction model alone may not be sufficient for analyzing some subtle differences in interaction with distinct environments. This is so because it does not take into consideration each environment's communicative characteristics. We suggest that further exploration of the communicative nature of each environment is required. A possible research goal is to devise a set of general pragmatic principles applied to interaction, and perhaps specialized to each particular environment.

Acknowledgments

Clarisse Sieckenius de Souza would like to thank CNPq for supporting her research. The authors would like to thank their colleagues at the Semiotic Engineering Research Group for providing valuable suggestions to this work.

References

- Brennan, S. "Conversation as Direct Manipulation" in B. Laurel (ed.) *The Art of Human-Computer Interaction*. Reading, MA: Addison-Wesley, 1990.
- de Souza, C.S. *The Semiotic Engineering of User Interface Languages*. *International Journal of Man-Machine Studies*. No. 39. pp. 753-773. 1993.
- Brown, G. and Yule, G. *Discourse Analysis*. Cambridge, UK: Cambridge University Press, 1983.
- Dahis, G. *Um Modelo Para a Especificação de Cenários de Interação* (in Portuguese). Master Dissertation. Informatics Department, PUC-Rio, Brazil, 2001.
- Grice, H.P. "Logic and Conversation" in (eds.) P.Cole & J.Morgan *Syntax and Semantics 3: Speech Acts*. New York, NY: Academic Press, 1975.
- Gross, B.J. & Sidner, C.L. "Attention, Intentions and the Structure of Discourse". In *Computational Linguistics*, vol.12, no.3, 1986.
- Leech, G. *Principles of Pragmatics*. New York, NY: Longman Inc., 1983.
- Wachtel, T. "Pragmatic Sensitivity in NL Interfaces and the Structure of Conversation". In *Proceedings of the 11th International Conference on Computational Linguistics*, 1986.

Avaliação da Navegação de Interfaces Web a partir de Modelos

Marco Winckler¹, Christelle Farenc¹, Philippe Palanque¹, Marcelo Pimenta²

¹LIHS – Université Toulouse 1
Place Anatole France,
31042 Toulouse, France

²Instituto de Informática – UFRGS
Caixa Postal: 15064
Porto Alegre - RS – Brasil
CEP 91501-970

{winckler, farenc, palanque}@univ-tlse1.fr, mpimenta@inf.ufrgs.br

Abstract. *Several models for accomplishing Web design and for supporting usability evaluation have been proposed in the last years. However, most of current usability evaluation methods don't take in account the interface specification to identify usability problems. In this paper we describe how we could identify usability problems by analyzing navigation model of Web interfaces. In addition, we discuss the advantages of usability evaluation with models over usability evaluation with prototypes.*

Keywords: avaliação de usabilidade, modelos navegação, verificação de modelos.

1. Introdução

Vários modelos para especificação de interfaces têm sido propostos nos últimos anos, especialmente para aplicações multimídia e baseadas em hipertexto, tais como XHMB [14], OOHDM [19], HMD [5], entre outros. Além disso, statecharts [9, 22] e redes de petri [20] têm sido explorados como modelos de base para representar navegação em hiperdocumentos. Tais modelos aportam um grau mais elevado de especificação e formalização para aplicações multimídia. No contexto de desenvolvimento para Web, extensões destes modelos têm sido propostas de maneira a contemplar características especiais de aplicações neste ambiente [1, 6]. No entanto, na prática, observa-se uma grande preferência por *storyboards* e *grafos* como ferramentas de modelagem [11, 17]. Em parte isto pode ser justificado pela simplicidade de utilização destes modelos e a excelente relação/custo benefício que eles apresentam [21], o que é compatível com a engenharia de usabilidade de baixo custo proposta por Nielsen [12].

Modelos de especificação são ferramentas úteis para o desenvolvimento de interfaces pelos mesmos motivos que eles são úteis no desenvolvimento de software em geral, pois permitem formalizar e especificar sem ambigüidade requisitos de interfaces e explorar possibilidades de implementação abstraindo a tecnologia utilizada. Além disso, o processo de modelagem em si é extremamente produtivo pois, ao especificar a interface, o designer é obrigado a analisar o problema em questão. Na prática, isto significa que uma série de erros podem ser eliminados ainda na modelagem.

No entanto, a simples utilização de modelos de especificação da interface não garante a sua usabilidade. Para identificar problemas de usabilidade existem vários métodos de avaliação que podem ser empregados ao longo do processo de desenvolvimento da aplicação. Atualmente, vários métodos de avaliação de usabilidade consideram requisitos específicos do ambiente Web [2, 8]. A grande maioria destes métodos identifica problemas a partir da observação direta ou indireta de usuários durante a utilização da interface [4, 10, 16]. Outros porém, como a avaliação por questionários [3], consideram informações indiretas sobre a utilização que podem sugerir a existência de problemas, como a insatisfação dos usuários com relação à interface. Outros ainda, como a avaliação heurística e métodos de inspeção [13], baseiam-se no conhecimento de um especialista que inspeciona a interface no intuito de identificar problemas de usabilidade. Em qualquer um dos casos, uma interface (ou um protótipo) é necessária para aplicação do método. Isto significa que, se utilizado um modelo para descrever a interface, este não é diretamente considerado durante a avaliação. Em alguns casos, o método ou ferramenta de avaliação de usabilidade cria um modelo da interface por engenharia reversa, ou seja, dada um aplicação existente, a ferramenta cria um modelo sobre o qual realiza a avaliação [15]. Observa-se, porém que este modelo gerado por engenharia reversa obviamente não é o mesmo usado para criar a aplicação e, desta forma, todos os componentes que são explícitos no processo de concepção não são considerados.

Os métodos de avaliação de usabilidade disponíveis para Web não aproveitam as informações contidas no modelo da interface para identificar problemas de usabilidade. Além disso, os resultados das avaliações de usabilidade, na maioria dos casos, precisam de um esforço de interpretação para serem aplicados ao design. Constata-se, assim, que existe um *gap* entre as etapas de design e avaliação de interfaces Web.

O objetivo principal deste trabalho é demonstrar como uma série de problemas de usabilidade podem ser identificados utilizando somente um modelo de especificação. Com isto, pretende-se estimular a utilização de modelos de concepção de sites Web e o desenvolvimento de ferramentas baseadas nestes modelos. Esta abordagem sugere a verificação automática ou semi-automática da usabilidade a partir de modelos de interfaces Web. Vários aspectos relacionados à interface podem ser avaliados através de modelos. Contudo, neste trabalho, somente o aspecto navegacional das interfaces é investigado. Optou-se pelo modelo StateWebCharts [23], que ressalta o aspecto navegacional, e que consideramos adequado aos propósitos deste artigo. Isto não significa que as verificações propostas sejam estritas a este modelo pois elas podem ser aplicadas a outros modelos de navegação.

Este artigo está organizado como segue. A seção 2 discute o ciclo de vida de desenvolvimento de aplicações para ambiente Web e requisitos para o processo de modelagem neste ambiente. A seção 3 descreve um estudo de caso de especificação de interface Web utilizando a notação StateWebCharts para o site do projeto Hidrologia da Bacia Amazônica. A seção 4 apresenta uma discussão sobre que aspectos de usabilidade podem ser avaliados sobre o modelo apresentado na seção precedente. Na seção 5 é discutida a análise realizada. E finalmente, na seção 6, são apresentados conclusões e trabalhos futuros.

2. Ciclo de Desenvolvimento de Aplicações Web

O ciclo de desenvolvimento de interfaces Web é uma espiral contínua, notadamente marcado por sucessivas modificações, que são muito mais frequentes em aplicações Web

que em outros tipos de interfaces. Dentro deste ciclo espiral, várias etapas se sucedem. O número e a importância de cada uma das etapas varia em função da abordagem utilizada. Neste trabalho, está sendo considerado o ciclo de desenvolvimento para interfaces Web proposto por Scapin et al [18] (ver figura 1), que compreende as seguintes etapas:

- Etapa 1:** Análise de requisitos: a estrutura do site e o contexto de utilização são identificados;
- Etapa 2:** Especificação: modelos da interface são construídos a partir dos requisitos obtidos durante a análise de requisitos;
- Etapa 3:** Projeto: modelos são refinados e o site é implementado de acordo com o seu conteúdo;
- Etapa 4:** Implementação: corresponde a criação de páginas HTML e objetos de som/imagem necessários à aplicação ;
- Etapa 5:** Utilização e avaliação: são avaliadas a usabilidade da interface e a coerência da interface com relação aos requisitos iniciais;
- Etapa 6:** Manutenção: envolve um ciclo de maior duração que envolve a coleta de novos requisitos e planejamento das modificações identificadas durante a etapa de avaliação.

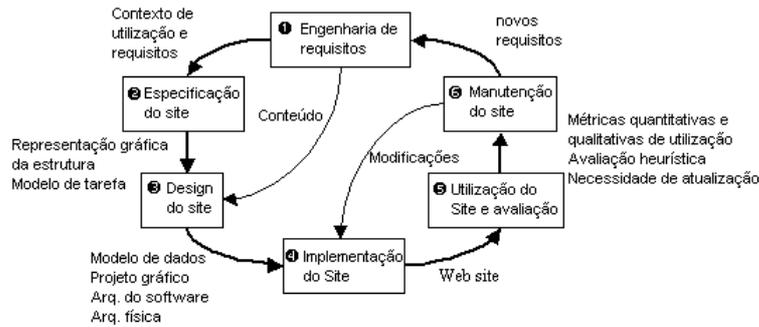


Figura 1 – Ciclo de desenvolvimento de aplicações Web [18].

Observa-se que, na figura 1, a prototipação não ocupa a parte central do ciclo mas é considerada como parte da etapa de manutenção. Um atalho possível neste ciclo permite a implementação logo após a análise de requisitos sem passar pela etapa de especificação, o que é freqüentemente observado na prática. Contudo, isto dificulta a construção de sites com maior usabilidade pois, devido a necessidade de modificações freqüentes no site, cada vez que um desenvolvedor altera manualmente uma interface ele está sujeito a incluir um problema de usabilidade. Estes problemas podem estar associados a um erro relacionado a uma operação em si (p.ex. um erro de digitação que torna um *link* inacessível) ou uma modificação na estrutura de navegação que elimina um caminho a um ramo da estrutura antes acessível.

A abordagem utilizada neste trabalho é evitar os atalhos e seguir todas as etapas do ciclo de desenvolvimento. Assim - sempre referenciando a fig. 1 - cada alteração da interface (novos requisitos como entrada da etapa 1) deve ser especificada (etapa 2) antes de executada sobre a interface (etapas 3 a 5). Embora a manutenção torne-se mais complexa e demorada, é possível verificar o modelo a cada alteração. Além disso, a especificação da interface documenta a evolução do site.

3. Especificação de Interfaces Web: Um Exemplo

Vários modelos para especificação de interfaces multimídia e baseada em hipertexto têm sido propostos nos últimos anos, tais como XHMBS [14], OOHDM [19], HMD [5], statecharts [9, 22] e redes de Petri [20]. Neste trabalho, optou-se pela notação StateWebCharts [23], que é uma extensão de Statechart [7] concebida especificamente para modelar interfaces Web. Entre outros recursos, esta notação representa visualmente os limites da aplicação modelada, controle de ações sobre a interface (usuário ou sistema), representação de distribuição cliente/servidor e composição visual da aplicação [23]. Estas características não são consideradas nos modelos mencionados e, por considerá-las importantes, optou-se por StateWebCharts.

Não é objetivo deste trabalho descrever toda a notação StateWebcharts, por isto, apenas os conceitos importantes para a discussão sobre a avaliação de usabilidade serão apresentados. Como exemplo, é apresentado uma modelagem parcial do site Web HIBAM: Hidrologia da Bacia AMazônica. O objetivo do site, que ainda está em desenvolvimento, é disponibilizar os resultados das pesquisas realizadas em cooperação Brasil-França neste tema durante a última década.

Antes de especificar a navegação com StateWebCharts, foi construída uma descrição hierárquica das informações do site, conforme figura 2. Observe que este modelo descreve instâncias (e não classes) de informações contidas no site e que, devido ao grande número de ramificações da árvore hierárquica, vários ramos não são detalhados. Os objetivos desta representação hierárquica são: a) criar uma lista completa das informações que fazem parte do site; e, b) organizar as informações em grupos de forma coerente ao domínio da informação.

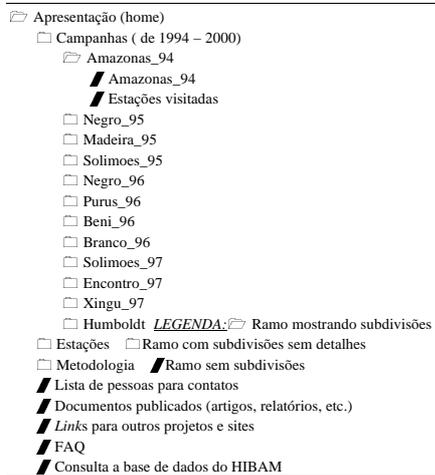


Figura 2. Modelo hierárquico de informações HIBAM.

Os itens representados na figura 2 foram utilizados para construir a especificação da navegação com StateWebCharts. A descrição hierárquica do site pode sugerir a criação de documentos do site, porém, a transposição da hierarquia lógica para documentos físicos

envolve outros aspectos como a complexidade da informação representada e a estrutura de apresentação que não são representados pelo modelo. Do mesmo modo, o modelo hierárquico pode auxiliar na identificação de *links* entre páginas, mas não representa todas as ligações. Salienta-se que o modelo navegacional não pode ser gerado automaticamente a partir do modelo hierárquico. O modelo apresentado pela figura 2 serve apenas como referência das informações que o modelo navegacional deve conter, ainda que a representação hierárquica possa inspirar o designer durante a especificação da navegação.

A idéia geral de StateWebCharts é que interfaces Web podem ser representadas por um conjunto de estados, sendo que cada estado representa os objetos da interface que o usuário poder ver e/ou interagir quando o estado está ativo. A concorrência em StateWebCharts tem a significação de concorrência visual e não concorrência de execução, como ocorre em statecharts. A semântica de um estado é, portanto, uma janela de visualização que contém uma lista de objetos a serem visualizados e, se dois estados são concorrentes, eles são visíveis simultaneamente. Fisicamente, cada lista de objetos terá um documento HTML associado onde transições entre estados são implementadas como *links* entre documentos. São consideradas como janelas: janelas do *browser*, cada *frame* (*frames HTML*) individualmente e caixas de diálogo. Assim, dado um estado do modelo, obtém-se o documento que ele contém e em que janela ele será visualizado.

A modelagem com StateWebCharts inicia com um estado de alto nível (nível zero) que representa a área de visualização principal do usuário e todos os sub-estados que a aplicação contém. Um estado é representado por um identificador (canto superior esquerdo) e uma descrição do seu conteúdo. Estados que contém outros sub-estados visualmente concorrentes, devem se detalhados em um nível hierárquico inferior. Por exemplo, a figura 3 apresenta um modelo parcial do site do HIBAM, representado os níveis 0 (figura 3.a) e 1 (figura 3.b). Observe que no nível zero o estado *S10* é representado sem nenhuma descrição, sendo que todo o seu conteúdo é apresentado em detalhe no nível 1 (figura 3.b).

Ferramentas como *bookmarks*, *history*, e acesso direto a *URLs* (*Uniform Resource Locator*) permitem que usuários iniciem a visita à aplicação Web por qualquer documento e são consideradas dentro do modelo para fins de avaliação. Contudo, o designer deve definir sempre um estado inicial (*default*) para aplicação. Estados *default* são marcados com uma seta iniciada por um círculo, como indica o evento *e1*. Este símbolo também é utilizado para indicar quando uma nova janela será aberta, como ocorre com a transição *e16*. Observe que na figura 3.b, o evento *e10* (entrada para o estado *S10*), é duplicado como entrada para os sub-estados *S11* e *S12* indicando, assim, que duas janelas concorrentes serão inicializadas.

Estados são representados conforme a função que exercem dentro do modelo. Estados que pertencem ao domínio da aplicação e, sobre os quais o designer tem poder de atualização, são representados como retângulos com cantos arredondados (ver *S1*, por exemplo), enquanto estados externos são representados sob a forma de nuvem (ver *S20*, por exemplo). Estados que são executados no lado servidor da aplicação Web são marcados em cinza (ver *S21*), sugerindo que eles não são estados visíveis pelo cliente. Quando o estado é gerado dinamicamente, este é representado com linhas tracejadas como ocorre com o estado *S22*, por exemplo. Com relação aos *links*, linhas contínuas indicam que o usuário tem controle sobre a transição (eventos *e1* e *e2*, por exemplo) e linhas tracejadas indicam

que o sistema pode causar uma transição automaticamente sem a intervenção do usuário (eventos $e20$ e $e21$).

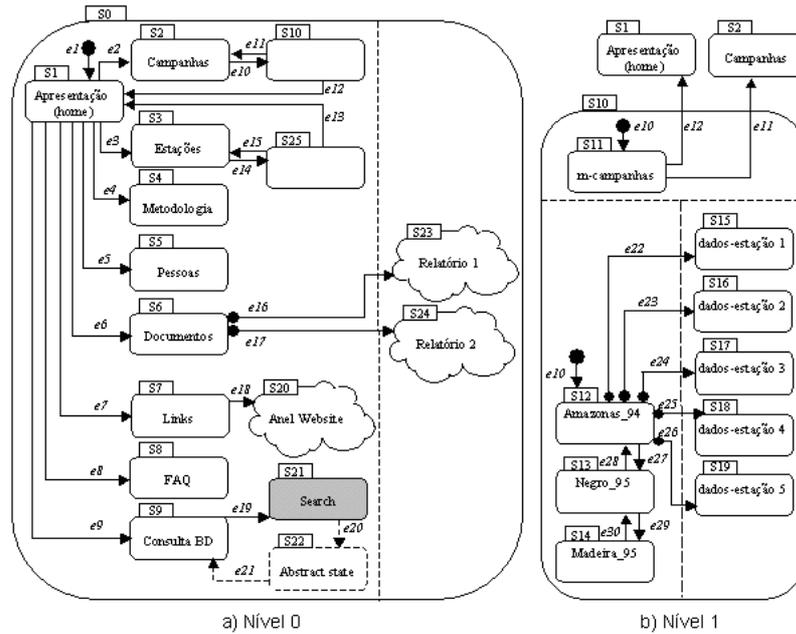


Figura 3 – Modelo de navegação do site HIBAM, níveis 0 e 1.

4. Verificação do Modelo de Navegação

Tendo como base uma especificação da interface em StateWebCharts pode-se realizar algumas verificações sobre o modelo que podem identificar ou sugerir a existência de problemas de usabilidade. Estas verificações não podem ser consideradas como métodos de avaliação automática, pois elas apenas ressaltam características do modelo que devem ser analisadas pelo designer a fim de determinar a ocorrência ou não de problemas de usabilidade.

Para as verificações são consideradas os seguintes dados que podem ser obtidos do modelo:

- A identificação do estado;
- Descrição do conteúdo que o estado armazena;
- Lista de eventos que saem do estado causando transições para outros estados;
- Quem tem o controle sobre a transições (usuário ou o sistema);
- Tipo de estado: externo, cliente, servidor ou transiente (gerado dinamicamente);
- A posição na hierarquia de estados, no caso de estados compostos;
- Concorrência visual dos estados;

Podem ser realizadas cinco classes diferentes de verificações de modelos, todas complementares entre si: i) análise estática do modelo; ii) análise do caminho navegado; iii) verificação de regras de comportamento; iv) comparação com a análise de tarefa; e v) determinação da equivalência entre modelos.

i) Análise estática do modelo

Este tipo de análise compreende um conjunto de verificações que podem ser feitas automaticamente sobre o modelo, sem considerar dados de simulação ou objetivo da navegação. Esta análise compreende cinco tipo de verificações:

- **Completude do modelo:** verifica se todos os itens descritos no modelo hierárquico de informação estão representados no modelo de navegação. As informações no modelo hierárquico representam requisitos para a aplicação e, se estes requisitos não fazem parte do modelo de navegação isto implica que o modelo pode estar incompleto ou impreciso.
- **Identificação de estados inacessíveis:** identifica estados que não tem ligações dentro do modelo e que, por consequência, são inacessíveis pelos usuários, o que pode ser considerado um problema de usabilidade.
- **Identificação de links para estados inválidos:** identificar links que não tem associações com estados válidos (*broken links*). Links inválidos podem ocorrer dentro do modelo quando, por exemplo, um estado é removido mas ainda existem referências no modelo. Se o estado é externo ao modelo, pode ser verificado a validade da URL indicada. Links inválidos podem ser considerados como problemas graves de usabilidade pois frustram o usuário e limitam a disponibilidade a um recurso da interface.
- **Identificação de links redundantes:** links redundantes podem ocorrer quando dois ou mais links são incluídos na mesma lista de objetos de um estado ou pela composição de estados concorrentes. Links redundantes podem ser tanto desejáveis, quando se quer múltiplas possibilidades de acesso a um recurso da interface como podem representar problemas de usabilidade quando são excessivamente redundantes ou confundem os usuários. Por isto, tal situação deve ser verificada.
- **Anomalia de composição visual:** pela associação de elementos concorrentes verifica-se a composição visual destes estados e eventuais anomalias. Como exemplo de anomalia cita-se a ocorrência de dois estados concorrentes que possuem o mesmo conteúdo. Outro tipo de anomalia que pode ocorrer é a existência de *looping* entre janelas (uma dentro da outra sucessivamente).

ii) Análise do caminho navegado

Teoricamente, a estrutura de navegação deveria dar ao usuário a possibilidade de explorar toda a interface de maneira flexível. Contudo, quando uma aplicação Web ultrapassa uma centena de páginas, torna-se, por vezes, difícil assegurar flexibilidade. A flexibilidade de navegação pode ser analisada pelo caminho navegado. O caminho é definido como a seqüência de estados que podem ser alcançados através dos links definidos em cada estado. Para fins de análise, qualquer estado dentro do modelo pode ser considerado como estado inicial. Dentro de modelos StateWebCharts, é possível isolar sub-estados de tal modo a analisar qualquer estado com relação a sua navegação rumo a outro estado independente da

sua posição na hierarquia. Para realizar este tipo de verificação é necessário indicar um estado inicial e um estado final. Com estes dados, pode-se verificar os seguintes cenários:

- **Existência de um caminho possível:** verifica se é possível atingir um estado-alvo a partir do estado atual. Esta simples verificação automática pode determinar problemas de usabilidade relacionados a não existência de um caminho para a informação.
- **Identificação do caminho mais curto:** verifica automaticamente todos os caminhos possíveis e, entre eles, o mais curto. Caminhos muito longos podem indicar problemas de usabilidade se conceitualmente os dois extremos são conceitualmente próximos. A indicação de caminhos longos dentro do modelo permite que o designer verifique se o esquema de navegação é coerente com os objetivos para a aplicação.
- **Identificação de caminho obrigatório:** a análise de caminho também pode revelar a existência de caminhos de passagem obrigatórios ao usuário. Esta pode ser uma estratégia interessante para o designer da aplicação para por exemplo "obrigar" o usuário a visualizar uma propaganda enquanto está navegando. No entanto, caminhos obrigatórios podem significar que a navegação não é flexível. A identificação de um caminho obrigatório em si, não indica a ocorrência de problema de usabilidade mas chama a atenção do designer para este tipo de construção.

iii) Verificação de regras de comportamento do sistema

Este tipo de verificação é feito sobre o modelo a partir de um conjunto de *regras de comportamento do sistema*. São denominadas regras de comportamento do sistema o conjunto de regras que podem ser aplicadas diretamente sobre o modelo e que definem um comportamento particular para a aplicação. No momento, o interesse é poder aplicar estas regras sobre o modelo de navegação. Estas regras podem ser descritas formalmente com o uso de Lógica Temporal [24]. Uma das vantagens do uso de StateWebCharts é que, por ser um modelo preciso, permite que regras descritas em Lógica Temporal sejam aplicadas ao modelo. Neste trabalho, nos limitamos a explicar conceitualmente como tais regras podem ser verificadas automaticamente, uma vez que descrevê-las formalmente esteja além do escopo deste artigo.

As regras podem ser construídas diretamente sobre instâncias do modelo ou a partir de dados genéricos. Por exemplo, considerando a regra geral: *“nenhum estado deve conter mais do que 7 links para outros estados”*, essa pode ser automaticamente verificada analisando todos os estados do modelo e contabilizando o número de *links* em cada um. Outro exemplo de regra que pode verificada automaticamente pelo mesmo princípio: *“todos os estados do site devem ter um link para a página principal”*. Outro exemplo: *“Todos os estados dentro do site devem ser acessíveis a partir da página principal através de, no máximo, 5 estados intermediários”*.

Regras de comportamento do modelo também podem ser obtidas a partir de cenários de uso ou por análise de tarefas e descrever comportamentos ainda mais específicos. Por exemplo, considerando a seguinte regra para o modelo HIBAM: *“todo usuário deve passar pela lista de campanhas para ter acesso a um campanha específica”*. Este é um tipo de regra construída sobre instâncias, ou seja, estados e eventos descritos explicitamente no modelo.

A grande vantagem da definição e uso de tais regras é que elas podem ser automaticamente verificadas. Isto é especialmente útil quando ocorrem modificações no site que podem alterar comportamentos de navegação que são considerados importantes. Por exemplo, considere que o modelo apresentado na figura 3 tenha sido modificado de tal forma a permitir o acesso direto a uma campanha em especial sem passar pelo estado *S2-Campanhas*. Neste caso, a verificação automática da regra acima identifica que a condição não é mais verdadeira dentro do novo modelo. Tal advertência poderá ser interpretada pelo designer como um problema de usabilidade ou não: esta verificação apenas visa mostrar ao designer quais comportamentos foram alterados dentro do modelo.

Deve-se salientar que regras de comportamento tem um campo de abrangência limitado e nem todos os tipos de comportamentos dentro de um modelo podem ser especificados por Lógica Temporal. Além disto, tais regras podem guiar o processo de concepção, mas novamente, é necessário a intervenção do designer ou de um avaliador para determinar se a violação de uma regra representa ou não um problema de usabilidade.

iv) Comparação com a análise de tarefa

Um tipo de verificação mais elaborada pode ser realizada semi-automaticamente a partir da análise da tarefa do usuário. Para explicar como isto pode ser feito, considere que existe um modelo de tarefa de usuário que descreve ordenadamente todos os passos que participam da realização de uma tarefa. A partir de um cenário de uso ou da análise de tarefa que gera esta descrição, pode-se verificar se existem estados no modelo que contemplam as subatividades; isto pode ser feito pela comparação de palavras-chaves contidas na descrição do estado e na descrição da tarefa. O próximo passo é verificar se a seqüência de atividades é compatível com a navegação fornecida pelo modelo.

Se não existe nenhuma compatibilidade entre a análise de tarefa realizada e o modelo, pode-se sugerir que existem problemas de usabilidade relacionados à ausência de informações/ funcionalidade, ou que o modelo suporta a realização da tarefa de uma maneira que não é compatível com a realizada normalmente pelo usuário.

A diferença entre este tipo de avaliação e a verificação de regras de comportamento do modelo é que a análise de tarefa é uma modelagem realizada sobre o trabalho real do usuário independente da interface utilizada, enquanto as regras de comportamento devem ser descritas sobre o modelo com o objetivo de garantir o seu funcionamento.

v) Determinação da equivalência entre modelos

Nesta verificação, avalia-se a equivalência de navegação entre modelos. Considera-se dois modelos equivalente se eles possuem o mesmo conjunto de caminhos de navegação. Para determinação de equivalência não são considerados a apresentação visual de um modelo (número de janelas e forma dos objetos contidos no estado), de tal maneira que dois modelos equivalentes com relação à navegação podem ser completamente distintos em aparência. Além disso, utilizando-se *regras de comportamento*, pode-se determinar a equivalência parcial de modelos que não correspondem a regra de equivalência total de navegação mas que obedecem as regras estabelecidas pela análise de tarefas.

Este tipo de verificação não tem um impacto direto sobre a identificação de problemas de usabilidade mas é útil quando várias estratégias de design são possíveis para uma mesma interface e deve-se escolher uma. Neste caso, a verificação de similaridade entre modelos de navegação pode orientar a decisão.

5. Discussão

As verificações apresentadas acima foram concebidas para serem realizadas com o auxílio de ferramentas automatizadas. Algumas das verificações como a *análise estática do modelo (i)* e *análise do caminho navegado (ii)* podem ser executadas manualmente sobre o modelo ou diretamente sobre uma interface, embora o esforço requerido seja considerável. Atualmente, existem algumas ferramentas que realizam, por exemplo, a *verificação de estados inacessíveis* ou *links para estados inválidos*. Contudo, estas ferramentas, que não utilizam nenhum modelo de especificação da interface, são limitadas a apenas esse tipo de avaliação simplificada. Verificações mais elaboradas, como por exemplo, a *verificação de regras de comportamento do sistema (iii)*, *comparação com a análise de tarefa (iv)* e *determinação da equivalência ente modelos (v)*, não são possíveis sem o auxílio de uma ferramenta e de um modelo.

Algumas das verificações, notadamente *i*, *ii* e *iii*, podem inicialmente ser empregadas durante as etapas de especificação e design do site (etapas 2 e 3, figura 1) como guias para a concepção e como verificações preventivas de potenciais problemas de usabilidade. Na etapa implementação do site e manutenção (etapa 4 e 6), tais verificações são úteis como forma de verificar a consistência das alterações com relação aos objetivos iniciais da interface. Na etapa de utilização do site e avaliação (etapa 5) todas as verificações propostas são consideradas métodos de avaliação da usabilidade.

O conjunto de verificações apresentado serve num primeiro momento como um guia ao designer auxiliando-o durante a concepção e depois como método de avaliação de usabilidade. Além disso, sugere-se que tais verificações sejam uma forma de aproximar as etapas de design e avaliação.

A maior parte das avaliações realizadas aqui utilizando a notação StateWebCharts pode ser reproduzida com outros modelos, tais como OOHDM, XHBMS, HMD e WEBML. Optou-se pela notação StateWebCharts por ser uma extensão de statechart, que é um modelo amplamente difundido e parte do conjunto UML. Uma vez que UML é um padrão atual para especificação de sistemas e bastante difundido entre desenvolvedores de aplicações Web, StateWebCharts torna-se transparente àqueles familiarizados com UML. Outro motivo pelo qual esta notação foi utilizada é que ela reforça o aspecto de concorrência visual de estados e representa claramente os componentes cliente e servidor da aplicação. No entanto, o objetivo principal não é a notação, uma vez que as verificações propostas são aplicáveis a outros modelos. O objetivo foi demonstrar que utilizando um modelo de especificação de navegações pode-se realizar uma série de avaliações que não seriam possíveis sem a descrição formal da interface.

Neste trabalho, somente o aspecto navegacional das interfaces foi considerado, o que na opinião dos autores não reduz a contribuição dos resultados: a maior parte dos problemas de usabilidade mencionados pelos usuários são exatamente problemas relacionadas à navegação e à organização da informação. Considera-se que esta mesma abordagem de avaliação baseada em modelos pode ser estendida para avaliar outros aspectos da interface, sendo este um tema a ser abordado futuramente.

As verificações propostas aqui não determinam propriamente a ocorrência de problemas de usabilidade relacionado à navegação mas apontam pontos críticos sobre o modelo que podem sugerir a existência potencial de problemas. Cabe salientar que este tipo de

avaliação não elimina a necessidade de avaliação de usabilidade por outros métodos pois a avaliação sobre o modelo não produz nenhum tipo de informação qualitativa ou subjetiva, que são essenciais para definir a usabilidade geral da interface. No entanto, é evidente que vários tipos de problemas podem ser determinados diretamente sobre o modelo o que evita a propagação de problemas em cascata até a implementação. Além disto, a avaliação de usabilidade sobre a interface pode ser aprofundada nos pontos em que a avaliação sobre o modelo não é suficiente. Esta abordagem acompanha a evolução da aplicação através do modelo de interface.

É importante notar que todas estas avaliações podem ser feitas diretamente sobre o modelo, antes mesmo que qualquer linha de código HTML tenha sido escrita. Contudo, tais avaliações são limitadas à usabilidade de navegação. Ainda assim, nem toda a navegação possível pode ser expressa em tempo de modelagem. Por exemplo, quando estados são gerados dinamicamente, novas ligações podem ser incluídas de maneira imprevisível.

Mesmo que modelos possam ser considerados complexos para a maioria das aplicações Web, considera-se que o retorno e suporte por eles oferecido deve compensar o esforço de modelagem. A identificação precoce de problemas de usabilidade associados à navegação é utilizada aqui como argumento para incentivar o uso de modelos. Contudo, a complexidade dos modelos pode ser simplificada através de métodos de visualização mais eficientes e de mais fácil utilização.

6. Conclusões e Trabalhos Futuros

É nossa convicção que problemas de usabilidade podem ser identificados ainda na etapa de design se modelos de especificação de interface são utilizados. Num primeiro momento a especificação auxilia na organização da estrutura da interface. Em seguida o modelo, formalmente descrito, pode ser verificado a fim de identificar possíveis focos que possam representar problemas de usabilidade. É importante ressaltar que todas as verificações aqui descritas são feitas sobre o modelo, e não sobre a interface. Na prática, isto significa uma economia de tempo e esforço da parte dos desenvolvedores, pois tais verificações podem ser feitas de maneira automática ou semi-automática guiando a avaliação sem a participação de usuários. Por outro lado, tais verificações podem evitar que usuário venham a ter contato com problemas de usabilidade que foram precocemente identificados e eliminados da aplicação.

Ferramentas que suportem a concepção de interfaces usando modelos são fundamentais para a realização da maioria das verificações propostas. Ainda que seja possível realizar algumas manualmente, isto seria difícil e tedioso. Assim, entre as perspectivas de continuidade deste trabalho está a construção de ferramentas que integrem as verificações apresentadas. Ainda é previsto que objetos de interação possam ser especificados dentro da notação StateWebCharts, o que poderá aumentar o número de verificações possíveis de maneira automática a partir da descrição da interface. Pretende-se também investigar exemplos com outras notações de especificação de interfaces Web e validação das verificações propostas.

Agradecimentos

Os autores agradecem ao CNPq pelo suporte à este trabalho e à equipe do projeto Hibam, em especial à Naziano Filizola pelo material e apoio para o estudo de caso.

REFERÊNCIAS

1. CERI, S.; FRATERALI, P.; BONGIO, A. Language (WebML): a modeling language for designing Web sites. 9th World Wide Web Conference, Amsterdam, May 15-19, 2000.
2. CHI, E. H.; PIROLI, P.; PITKOW, J. The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site. Pages 161-168. Proceedings of the CHI 2000. April 1 - 6, 2000, The Hague Netherlands.
3. CHIN, P. J.; DIEHL, A.; NORMAN, K. Development of a tool measuring user satisfaction of the human-computer interface. Proceedings of CHI88, p. 213-218.
4. FULLER, R.; DE GRAAFF, J. J. Measuring User Motivation from Server Log Files. Proc. of 2d Conf. on Human Factors and the Web, October 1996.
5. GARZOTTO, F.; PAOLINI, P.; SCHWABE, D. HDM: a model-based approach to hypertext application design ACM Transactions on Information Systems Vol. 11, No. 1 (Jan. 1993), p. 1-26.
6. GÜELL, N.; SCHWABE, D.; VILAIN, P. Modeling Interactions and Navigation in Web Applications. Proceedings of WWW and Conceptual Modeling 2000 Workshop; Salt Lake City, USA. 2000.
7. HAREL, D. Statecharts: a visual formalism for complex systems. Science of Computer Programming 8, P. 231-274. 1987.
8. HARTSON, H. R. et. al Remote Evaluation: the network as an extension of Usability Laboratory. Proceedings of CHI96, ACM Press, Nova Iorque, 1996, p. 228-235.
9. HORROCKS, I. Constructing the User Interface with Statecharts. Addison-Wesley, 1999.
10. IVORY, M.; HEARST, M. State of the Art in Automated Usability Evaluation of User Interfaces. Submitted for publication, 2000. Disponível por WWW em: <http://www.cs.berkeley.edu/~ivory/research/web/papers/survey.pdf>
11. LYNCH, P. J.; HORTON, S. Web Style Guide: Basic Design Principles for Creating Web Sites. Yale Univ Press, 1998.
12. NIELSEN, J. Usability Engineering. Boston - USA: Academic Press, INC., 1993. 362p.
13. NIELSEN, J.; MACK, R. Usability Inspections Methods. New York: John Wiley, 1994. p.
14. PAULO, F. B.; TURINE, M.; OLIVEIRA, M.; MASIERO, P. C. XHMBS: a formal model to support hypermedia specification. Proceedings of the ninth ACM conference on Hypertext and hypermedia. June 20 - 24, 1998, Pittsburgh, PA USA. Pages 161-170.
15. RICCA, F.; TONELLA, P. Analysis and Testing of Web Applications, Proc. of ICSE'2001, International Conference on Software Engineering, Toronto, Ontario, Canada, May 12-19, 2001.

16. RUBIN, J. Handbook of Usability Testing: How to plan, Design, and Conduct Effective Tests. New York : John Wiley, 1994. 330 p.
17. SANO, D. Designing large-scale web sites: a visual design methodology. John Wiley & Sons, New York. 1996. 288 P.
18. SCAPIN, D.; et al. Transferring Knowledge of User Interfaces Guidelines to the Web. In. Tools for Working with Guidelines. London: Springer; 2001.
19. SCHWABE, D.; ROSSI, G. Developing Hypermedia Applications using OOHDM, Workshop on Hypermedia Development Processes, Methods and Models, Hypertext'98, Pittsburgh, USA. 1998.
20. STOTTS, D. P.; FURUTA, R. Petri-net-based hypertext: document structure with browsing semantics. ACM Trans. Inf. Syst. 7, 1 (Jan. 1989), pp. 3-29.
21. VIRZI, R. A.; SOKOLOV, J. L.; KARIS, D. Usability Problem Identification Using Both Low- and High-Fidelity Prototype. Proceedings of CHI'96 Human Factors in Computing Systems, April 13 - 18, 1996, Vancouver Canada.
22. ZHENG, Y.; PONG, M. Using statecharts to model hypertext. In. Proceedings of the ACM conference on Hypertext. Milan, Italy. 1992.
23. WINCKLER, M.; FARENC, C.; PALANQUE, P. BASTIDE, R. Designing Navigation for Web Interfaces. Submetido para publicação à conferência IHM-HCI2000, Lille, França. 2001.
24. EMERSON, E. A.; SRINIVASAN, J. Branching Time Temporal Logic. In.: Lecture Notes in Computer Science (354) - Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency. Spring-Verlag, Berlin. 1988.

Bringing Interface Design and Software Development Processes Together: How Organizational Issues Impact Product's Usability

M. Cecília C. Baranauskas¹ Juliana Salles² and Roberto S. Bigonha²

¹Institute of Computing- IC
State University of Campinas – Unicamp
Cidade Universitária Zeferino Vaz
13083-970 Campinas – SP – Brazil

²Computer Science Department-ICEX
Federal University of Minas Gerais
Av. Antônio Carlos 6627
31270-010 Belo Horizonte - MG – Brazil

cecilia@ic.unicamp.br , {jusalles, bigonha}@dcc.ufmg.br

Abstract. *While usability issues have been widely stressed in literature as a goal of the software design methodologies, quality in the design and development process has not received the same attention. This work aims to investigate software production processes by addressing the communication among work groups in a software design organization. Our focus is on understanding the communication process that takes place between groups in a software design organisation, the nature of the changed messages in relation to their meaning for the audience, and how they impact and are propagated in the whole process. We based our understanding of communication in Semiotic foundations, to propose a meta-model for analysing the whole process of system design and development. A case study with the design process of a large software organisation illustrates our proposal.*

1. Introduction

Organizational issues involved in software development and use have long been the focus of the Information Systems (IS) field. Nevertheless, some approaches in the Human-Computer Interaction (HCI) field have also concerns about organizational aspects influencing the design and use of computational systems. HCI and information systems developments have evolved from different backgrounds and focus to a common end: to achieve high quality in software usage [7]. While quality in the product has been widely stressed in literature as a goal of the software design methodologies, quality in the design and development process has not received the same attention. This work aims to investigate software production processes by addressing the communication among work groups involved in a software production organization.

A group can be defined as a collection of persons arranged or classified together because of some specificity, common interest, shared ideals, etc. A work group adds to this definition a shared goal and articulated actions to achieve it. This means that the interaction and relationship among the group members is an important differential. In the same way, work

organisations can be thought of as “*groups of groups*” [10], as the dynamic processes defining relationships among the groups are important factors in the promotion of the established goals.

Even though the organisation as a whole may have a single clear set of goals and interests, individuals and groups within the organisation may not share these goals and interests in the same way. For instance, in a software development organisation, members of the Usability Engineering team may differ substantially from members of the Development team in terms of their background and work practice. To most software engineers, the system is thought of in terms of its structure, functions, and components. To a usability engineer, the system includes the human user and all his potential actions interacting with the software. In addition, there are different performance parameters for different groups inside the organisation. While the software developers have their concern on development schedules and correction of code, the usability engineers have the preoccupation of insuring usability without causing delay in the process. A kind of tension between groups is not rare. In addition, within organisations, work groups can be influenced by the different cultures revealed through their assumptions, beliefs, self-image, feelings and fears, and the language they use to talk about their daily work [3], [10].

A comprehensive and integrated understanding of the connections among design and development activities is necessary at all levels of the process. Understanding and integrating the whole development process requires an understanding of how different groups in the organisation acquire and communicate this overview. As pointed out by Hix and Hartson [11] this seems to be true in any large system (e.g. a new automobile design). Participatory [15] and Contextual Design [3] propose several techniques as communication mechanisms in order to bring a design team to a shared understanding of the customer.

While we recognise the efforts of some methodologies for a shared understanding of the subject of design, we argue that a global view of the communication among the different groups in the organisation is important to ensure quality in the process and in the product, as a consequence. As a responsibility of the whole project, not of individual initiatives, new communication requirements emerge in the design and development process.

Semiotics allows us to understand information and communication aspects involved in organisations. Liu [13 p.7] points to “*a set of methods that can be used by researchers and business users in their understanding, development, management and use of information systems*”. Our focus in this work is on understanding the communication process that takes place between groups in a software design organisation, the nature of the changed messages in relation to their meaning for the audience, and how they impact and are propagated in the whole process. We based our understanding of communication on Semiotic foundations, to propose a meta-model for analysing the whole process of system design and development. A case study with the process of a large software organisation illustrated our proposal.

The paper is organised as follows: Section 2 presents the theoretical rationale for our understanding of communication in the design process. Section 3 presents a case study to highlight some results of representing communication involved in a product development cycle and walking-through the proposed meta-model. Section 4 concludes pointing out to further work.

2. Understanding Communication in the Software Design Process

In a software development organisation, all the work groups need to know what is going on, and the impact that the interaction and communication processes taking place among all these people has in the product.

“There are the rest of the engineers on the project who have to believe in the system enough to code it. There are the project teams working on systems that have to interface to yours. There is your manager, his manager (...) There is the marketing and the product-planning department who tend to be sceptical of ideas coming out of engineering. There is the sales force of a commercial product, which needs to understand what makes the new product worth selling. And there are the customers, who need to be convinced that the new system will improve their lives” [3, p. 199].

Beyer and Holtzblatt [3] argue that it is not feasible to include all the people who care about the result in the same design team. On the other hand, they maintain that a cross-functional team, despite being important to bring the perspectives of the different groups to design, does not guarantee communication back to the groups. The authors propose multiple strategies and techniques to enable communication among the groups, through the Contextual Design.

Contextual Design presupposes that any system embodies a way of working. As so, it is a method that define a series of actions that lead a team to agreement on what their customers need and how to design a system for them. The main contribution of this approach is a series of techniques that forces people to interact with data, to share a space for analysing customer data, to engage with work models, and mainly to tailor conversation to the concerns and work style of each involved group.

Semiotic approaches perceive the software interface as a communication act between designers and users, using the computer as medium [16], [6], [2]. The designers establish the limits of this communication and create a set of signs that users can activate [2], which means that designers are the senders of this communication model. Semiotic Engineering [6] considers the message itself a meta-communication artefact, since the interface exchanges messages with users.

The focus of this work is the communication involved in the design and development of computational artefacts. One aspect of this communication is the dialogue between designers and users. As Adler and Winograd point out [1], this kind of communication is embedded in every kind of artefact. Through their structure and appearance, designed objects express more or less effectively what they are, how they are used, and how they are integrated with the embedded context. According to semiotic approaches in HCI, this structure and appearance - i.e., the interface- can be understood as a message sent from designers to users using the computer as channel.

We argue that in order to design this message, all the involved groups need to develop a common understanding about what is going to be done. On the one hand, there are the users' needs and expectations; on the other hand, there are contextual, technological, budget and schedule constraints that limit the possible design solutions. Not only designers and users, but also the remaining agents must engage in a process of negotiation, which we understand as a conversational process. In this conversation, many communication acts occur, many messages are exchanged using different channels.

In this work, we argue that to understand the dimensions involved in the construction of the interface as message, it is important to develop a better understanding about the dialogue that occurs among the many parties involved in the design activity.

3. Design as a Matter of Communication

Design has been understood as a technical term referring to a software production step in which the lines of code are put together according to some software development methodology. In this work, as used by some influential authors in the area of HCI [27],[19],[24], the English language usage is adopted for the term, meaning the invention and organization of any structure.

In order to design the message (i.e. the interface), designers and users must engage in a dialogue using different channels. As Gould and Lewis point out [8], "*getting useful design information from prospective users is not just a matter of asking*". Designers need to ask questions, represent the information they get and the solutions they provide in a meaningful way to check it, perceive users' questions and reactions and so on [18], [20]. The designer-user dialogue has been carried out through usability tests in the successive phases of a design-implement-evaluate loop that characterizes the design process, which is, by its very nature, iterative.

More recently, the importance of the work context in the design process was widely recognised. Besides the evaluation of the users' performances when they interact in isolation with an artefact, another goal came onto the scene: designers realised that it was also important to analyse how an interface can reflect users' work context and support users' social interactions to execute their tasks. New approaches, like participatory [15] and contextual design [3], were developed and introduced additional channels in user-designer communication.

Besides designer-user communication, the other groups are also engaged in some type of communication through different channels: designers talk to marketing people, customer support mediates between developers and users, external consultants help both users' and developers' organizations, etc. The use of different channels is related to the needs of the communication acts: the nature of information desired, the projects' contexts, the audience's language, etc.

Communication Models and the computer as media

To understand communication in the design process, we must first form a coherent understanding of what communication itself involves. Several models for communication have been presented and discussed by thinkers from diverse philosophical schools. We will examine some of them in order to reach the essential aspects of the concept.

For a long time, the dominant model in the theory of communication was Shannon and Weaver's Mathematical Theory of Communication [25]. Grounded in the Theory of Information as a system of mathematical basis to study the problems of transmitting messages through physical channels, it is therefore a technique of communication engineering in which they identify source, encoder, channel, noise, decoder and receiver of information, as illustrated by Figure 1.

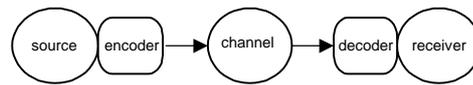


Figure 1. Shannon and Weaver's communication model

To understand the linguistic processes of verbal communication, Jakobson [12] proposed a model in which he identifies 6 factors that constitute every verbal communication act. The addresser sends a message to the addressee. To be effective, the message requires a context to which it refers, apprehensible by the addressee, a code total or partially common to both (addresser and addressee), and a contact, a physical channel between the addresser and the addressee. As in Jakobson's proposal, the semiotically improved model adds the role of the repertory of signs of both the sender and the receiver and requires the overlapping of the two repertories as a necessary condition for communication [5].

Westley and MacLehan [26] proposed a model derived from mass communication that reflects the various interactions among the entities (sender, receiver and channel), involved in the communication, including the receiver-sender, receiver-channel and channel-sender directions. In this model, the receiver plays an active role. He or she asks for information and it is this question that determines the content and form of the message created by the sender. There are feedback points that seem to better represent the conversational process; through them, the senders can add content to a message or change its form to facilitate the receiver's comprehension.

The direct transposition of the models derived from the Information Theory to understand human communication has many drawbacks, however. Parameters such as entropy or redundancy, while pertinent to "messages" are not the most important and do not reflect the specific nature of human communication. Another difficulty pointed out in the literature [5] is the unidirectional movement associated to the communication act, from a source (or addresser) to a receiver (or addressee). Receivers and senders actually engage in dialogues involving a process of meaning negotiation towards a common understanding. What receivers have understood from what a sender has "said" is frequently revealed in what receivers say themselves when they next take a turn at the communication act [4]. Moreover, these models do not explore alternative channels through which the communication between senders and receivers also take place. In our view the diversity of channels used for communication is what makes it inherently complex and powerful. On the other hand, models derived from verbal communication do not bring into focus the channel particularities and the ways they can shape the message being conveyed.

In designing the interface (i.e., the message), designers need that users express their needs and expectations, describe their tasks, work process, and so on. They exchange ideas with designers, present documents describing internal process to designers, show the designers the codes they use to represent their data, and so on. In doing so, users are communicating with designers using channels other than the computer artifact alone, so the designers can understand what needs to be designed. On the other hand, designers must check what they understood against what users have "said"; they must represent it in a meaningful way so they can "talk back" to users about it. They are also using channels to exchange these other messages; frequently, they are even creating new codes to carry out these communication acts.

As pointed out by Beyer and Holtzblatt [3], all artifacts even the most informal, have structure and content (information that the artifact carries). In designing the artifact, the designer establishes a conversation with the design medium, in the sense explained by Schon [24]. This level of communication is also part of the process and thus should be considered.

A fractal communication model, proposed in [21], [22] captures the nature of the communication process involved in software design, while stresses the fact that, in order to design the primary message (the interface), other fractionated messages must carefully be designed and appropriate channels must be chosen to convey them. Figure 2 illustrates the main concepts of the fractal model of communication.

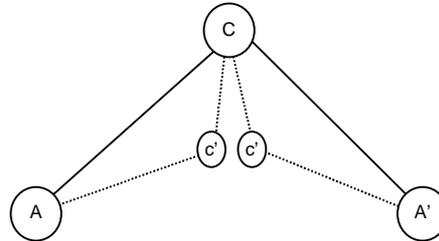


Figure 2: The fractal model of communication

In this diagram, nodes represent the communication agents (A and A') and channels (C and C'). The arrows are bi-directional, which means that an agent sends and receives messages. Nodes C' represent the fractal nature of communication. Different foci of the design process can be highlighted: the designer-user communication (A-A') using the interface as message, in a first level, as the interface is the unity- message conveyed by the computer (which is the first channel). The designer-artefact communication (A-C), and the user-computer communication (C-A') are represented in a second level of the fractal, having C' as special channels.

An inner level could be opened for analysis, showing a third level of the fractal structure, as illustrated by Figure 3. Usability engineers, for example, communicate with users using a usability test as channel. In designing the test, usability engineers communicate with the emergent artefact (the test) through a checklist as channel, for example.

In this model, the interface is understood as a unity-message that reflects what was grasped through the fractionated messages. So, the interface as a unity-message is directly affected by the choice of channels and messages used to compose the fractionated messages during the design process. This means that, in designing the interface, or the unity message, many fractionated messages are being exchanged. Each one of those messages is also being designed and should be carefully designed to make the designer-user communication through the unity-message smoother. As both senders and receivers in this communication process, designers and users change turns and cooperate on the development of the interface.

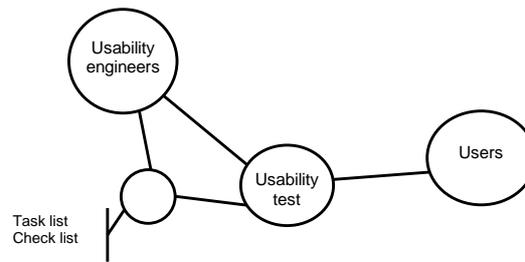


Figure 3. An instance of the fractal inner level

This model reflects the authors' understandings about design. This structure can better reflect the potential contributions of each part in the whole process. Also, in designing the interface, designers should also design other messages and choose appropriate channels to send them. The design of the fractionated messages is highly influenced by the development context [9] and the particular characteristics of each project. The conversational process (first level) and the designers and users communication with the emerging artefact (second level) are made explicit in the proposed model as we consider them fundamental aspects to improve the quality of the communication in the first level.

4. Meta-Modelling a Product Development Cycle: A Case Study

The meta-model was applied in a major software design company, given the pseudonym here "ORG", which employs leading professionals in all areas involved with the creation of commercial applications also called off-the-shelf software. The main groups identified as agents of the communication process were marketing people, program managers, designers, usability engineers, developers, testers, and support. Customers, clients, beta-users and users are four types of external agents called "users": clients are people who have already bought the product and receive the service of the support team for problems with the product. Customers represent the market: people who are identified by the marketing as a potential client. Users are the subjects recruited by the usability team to participate in the usability tests. Actually, these three categories of persons are potential end-users for the product.

Figure 3 illustrates the meta-model applied to the product development cycle of this particular organisation. An overview of the communication among the groups is showed in a simplified way, with two levels of the fractal structure.

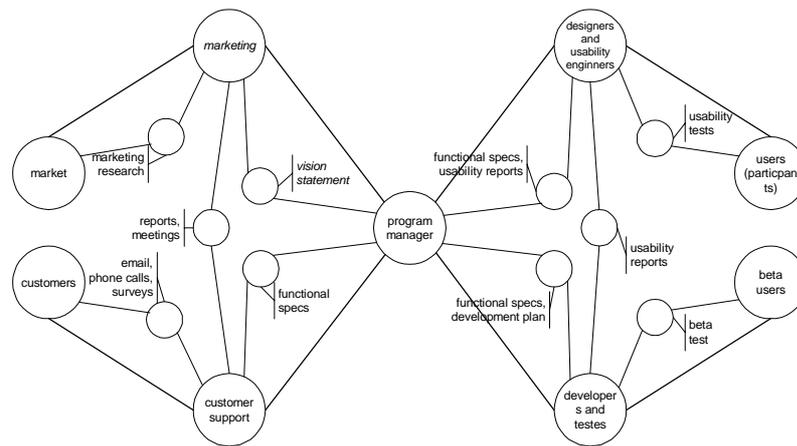


Figure 3. Meta-model for the product development cycle at ORG.

Marketers are responsible for ensuring a product meets a need for which people will pay money. Besides dealing with issues related to the market, they are engaged in product planning activities, through the product manager. They focus in demographics rather than on work practice, in identifying what kind of customer makes up the market. Marketers deal with feature lists, lists of customer needs, requirement lists, wish lists, etc. Customers are people who make the market and are targeted by the marketing group.

Program managers are the people in charge of the project, whose primary responsibility is to ensure the promised features are being completed on time and with acceptable quality.

Developers are programming engineers and have their focus on technology and on what make a clean program code. Traditionally they get directions from the marketing, through the program manager, who specifies what the overall system must do. They focus on clean implementation of the system design. Testers are people who work very close to the developers, testing code and reporting bugs.

Designers and usability engineers are the people who would have to be in charge of the product design, assuring usability for it. The usability group is a kind of quality control group, as they have the direct observation of the problems experienced by a user interacting with the system. Actually, at ORG designers and usability people are in charge of the product interface design and test.

Support are people in charge of responding to problems encountered by clients in the use of the system.

Typically the life cycle of the ORG products has three phases: planning, development and stabilisation phases. In the planning phase the functional specification and the project schedule is produced. In the development phase, internal liberation of a functionality subset is produced. In the phase of stabilisation, the product is extensively tested, in house and externally (beta tests).

During the planning phase, communication between the product manager (marketing) and the programming manager is established to produce the vision document. This artefact defines a set of goals that drive the product development. These agents also produce a high level specification for the product with a preliminary list of features to be present in the final product. The product life cycle at ORG is oriented by this vision document and by the high level specification.

During the development phase, communication between program managers and developers take place and the functional specification evolves as a result of this conversation. Developers have an important role in the choice of the features to be implemented. Subsets of this functionality are defined and milestones are stated for their liberation. Each set of features liberation involves intensive communication between developers and testers. While developers write code based on functional specification, the later test it for bugs. It is also in this phase that the interface is designed and tested for usability problems. After the last liberation, modifications in the main interface components (menus, dialogue boxes, etc.) are not allowed anymore.

During the stabilisation phase, the software product is extensively tested at ORG and outside, with Beta-version clients. Testers and developers establish communication with Beta-version clients during this phase. When high severity bugs are not found, the product is liberated to manufacturing.

The different agents also establish conversation with the emergent artefact, using channels related to their function in the organisation. As examples, there is a conversation between usability engineers and the emergent interface, using inspection methods as channels; developers establish a conversation with the code they are creating, through the tools they use (compilers, debugging tools, etc.).

Walking through the meta-model

An overview analysis of the meta-model for the design process point to aspects of communication among the agents, which deserve further discussion:

- The analysis on the meta-model tells us that the process is marketing-oriented, which is consistent with the economical and cultural contexts of ORG. This is reflected in the development driven by the functionality. The program manager acts as a channel between the marketing and all the other groups (developers, testers, designers and usability engineers). He propagates information to developers and testers using the functional specification, communicating the intended functions for the product and the priorities to implement them. From the developer perspective, this is an efficient message. However, designers and usability engineers need more elements to understand how the functions should be arranged and structured in the product as a whole, determining the user experience. On the other hand, designers and usability engineers' vision should be considered especially in the planning phase of the cycle. However, as the meta-model shows, their participation is noticed only in development activities (concentrated in the right part of the diagram), when the product specifications are already defined.
- The understanding about the user has different meanings to different agents of the process. The meta-model shows us different "users" communicating to different agents: some users participate in the usability tests, communicating to the usability

engineers; customers are the people the marketing is targeted to; clients are users in communication with the support group; beta-test participants are users communicating to testers. Nevertheless, the marketing group is the only agent involved in the identification of (**who** are) the prospective users. Other agents in the process demand understanding about the users: designers should know for whom they are designing. Marketers use specific channels (market research) to get a demographic description of the prospective users. However, the data interesting to the marketing group do not necessarily interests to the designers group. New channels and specific messages could be introduced between the agents in communication with those users and the designers group to convey a more complete view of the user and to accommodate specific necessities of designers and usability engineers.

- Support to the clients and beta-tests would be powerful additional channels to identify (**what** are) the usability problems, as they have direct access to problems pointed by users. However, in the first case, the data collected by the support group do not propagate to the usability engineers. In the beta-test case, the collected data are not adequate to answer for usability questions. Despite having the channel for communication, new messages should be designed in the beta-test to address also usability issues. Using data from beta-tests and from the support records would mean the utilisation of already existent opportunities. Data collected through these channels could be used as input in the design of usability tests to answer how the problems occur. Thus, the whole set of messages would be complemented more effectively.
- Certain channels, despite present, do not establish a psychological connection between two groups. In analysing a channel, it is important to adequate the messages to the audience. One group needs ways to communicate that are tailored to the concerns and work style of the other group. The program manager acts as a channel between the marketing and designers/usability engineers using the functional specification document. While developers think in functions as units of implementation (consistent with marketing objectives), designers need a much more wide view that is not communicated by the specification document. Also, the communication between usability engineers and developers through the usability report and email does not seem adequate to the developers who are much more technically focused. As a consequence this message does not reach the developer.

Results of a first analysis on the meta-model point to the necessity of new channels and messages to enhance communication among key elements of the process. The dynamic enabled by new communication channels and the message propagation could lead to a more integrated design and development process and potentially a better product.

5. Conclusion

In this work we presented our understanding of design as a communication process and proposed a meta-model that accommodates all the activities of the design/development process. While being a general model for communication, it captures the necessary contributions that should occur among the different agents of communication, as senders and receivers in this communication process. Also, it stresses the fact that, in order to design the

final interface, designers must carefully design other messages and choose appropriate channels to convey them, always considering the projects' resources and limitations.

In summary, the fractal communication model organises an analysis space which unifies some current independent practices of design while brings new issues which deserve more investigation to light. Several questions can be addressed with support of this meta-model, as for example: Are there enough channels and messages to answer to usability questions? (who are the users, what are the usability problems, how and why they occur)? Do the messages relative to each question propagate efficiently to the project team? Is there a type of communication act best suited for the design of a specific artefact? What are the consequences of a "broken" propagation resulting from lack of appropriate channel or message?

A Communicative Walkthrough - an inspection method based on the fractal model of communication is being proposed [23] for systematic analysis of the effectiveness of communicating usability issues among the groups involved with the product design and development. Further work is being done through case studies to evaluate how different design approaches fit into the proposed meta-model. The possibility of making explicit the communication levels stressed by the application of a particular technique or methodology brings new possibilities of analysing all the elements involved in the design of computer artefacts.

The context for developing off-the shelf products, as is the case at ORG, constrains, up to some extent, the interaction with users. With the design of new channels and messages to collect and to propagate data about what the usability problems are, usability engineers could make the communication with users much more effective. This view of the organisational context of software development allows to search for continuous improvement in the process and potentially to develop more usable and useful software.

Acknowledgements

This work was partially supported by the CNPq and FAPESP.

6. REFERENCES

1. Adler, P S, and Winograd, T A (1992) (eds.) *Usability: turning technologies into tools*, Oxford University Press, 1992.
2. Andersen, P B, (1997) *A Theory of Computer Semiotics*, Cambridge University Press, Cambridge, UK.
3. Beyer, H B and Holtzblatt, K, (1998) *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann Publishers.
4. Brown, G, (1995) *Speakers, listeners and Communication*, Cambridge University Press, Cambridge, UK..
5. Coelho Neto, J T,(1996) *Semiótica, informação e comunicação*, 4ª ed., Editora Perspectiva, SP Brazil.
6. de Souza, C S, (1993) The semiotic engineering of user interface languages, *International Journal of Man-Machine Studies*, v. 39, p. 753-773.
7. Ehn, P., Lowgren, J. (1997) Design for Quality-in-use: Human-Computer Interaction meets Information Systems Development, in: Helander et al (eds.) *Handbook of Human-Computer Interaction*, Elsevier Science.

8. Gould, J, and Lewis, C, (1985) Design for Usability: Key Principles and What Designers Think. *CACM*, pp. 300-311, v. 28, n. 3.
9. Grudin, J, (1991) Interactive systems: bridging the gaps between developers and users. *IEEE Computer*, April.
10. Grudin, J., Markus, M.L. (1997) Organizational Issues in Development and Implementation of Interactive Systems, in: Helander et al (eds.) *Handbook of Human-Computer Interaction*. Elsevier Science.
11. Hix, D., Hartson, H.R.(1993) *Developing user interfaces: ensuring usability through product and process*. John Wiley & Sons.
12. Jakobson, R. (1960) Closing Statement: Linguistics and Poetics, in: Sebeok, A, ed., *Style in Language* Cambridge, Mass.: The MIT Press.
13. Liu, K. (2000) *Semiotics in Information Systems Engineering*, Cambridge University Press
14. McLuhan, M, (1964) *Os meios de comunicação como extensões do homem*, Cultrix, SP Brazil.
15. Muller, M, (1997) Participatory Practices in the Software Lifecycle, in: Helander et al (eds.) *Handbook of Human-Computer Interaction*. Elsevier Science.
16. Nadin, M.(1988) Interface Design and Evaluation – Semiotic Implications, in: Hartson, H. R and Hix, D, (eds.), *Advances in Human-Computer Interaction*. v. 2. Ablex Publishing Corporation, 1988.
17. Norman, D A, (1988) *The Psychology of Everyday Things*, Basic Books.
18. Norman, D A, (1986) Cognitive Engineering, in: Norman, D and Draper, S W, ed., *User centered system design: new perspectives on human-computer interaction* Lawrence Erlbaum Associates.
19. Norman, D.A. (1998) *The Invisible Computer*, The MIT Press, Cambridge, Massachusetts
20. Rheinfrank, J, Hartman, W R. and Wasserman, A, Design for Usability: Crafting a Strategy for the Design of a New Generation of Xerox Copiers, in: Adler, P S and Winograd, T A, (eds.), *Usability: turning technologies into tools*, Oxford University Press.
21. Salles, J.P. (2001) O Modelo Fractal de Comunicação: Criando um Espaço de análise para Inspeção do Processo de Design de Software. PhD Thesis, May, 2001.
22. Salles, J.P., Baranauskas, M.C.C., Bigonha, R.S. (2000) A Communication Model for the Interface Design Process. Workshop in Semiotics, CHI2000
23. Salles, J.P., Baranauskas, M.C.C., Bigonha, R.S. (2001) Towards a Communication Model Applied to the Interface Design Process (forthcoming)
24. Schön, D and Bennet, J, (1996) Reflective Conversation with Materials, in: Winograd, T, (ed.), *Bringing Design to Software* ACM Press.
25. Shannon, C L and Weaver, W, (1949) *The mathematical Theory of Communication*, Urbana, Univ. of Illinois Press.
26. Westley B H and MacLehan, M S, (1972) Un modelo conceptual para la investigación en comunicaciones, in Smith G, (ed.), *Comunicación y cultura Nueva Visión*, Buenos Aires.
27. Winograd, T. (1996) *Bringing Design to Software* ACM Press.

Comparing usability evaluation methods applied to corporate web portals

Cláudia Dias

Depto de Ciência da Informação e Documentação - Universidade de Brasília
AOS 04 Bloco A apt 522 - Brasília - 70660-041, Brazil

claudiaad@yahoo.com

Abstract. *The variety of usability evaluation methods, the need to adapt them to the system's context of use, and the technological evolution of information management - the corporate web portal - suggested the research problem of comparing usability evaluation methods, applied to corporate portals' context of use. The goal of the present study was first to compare four usability evaluation methods, when employed by non-expert people, and second to verify the influence of the evaluators' experience and abilities on the results of usability evaluations of web portals.*

Keywords: *usability evaluation, corporate portal, web interface, human-computer interaction*

Introduction

In order to compare usability evaluation methods applied to corporate web portals, the context of use of the sample portal, selected for this study, was first analyzed, identifying its typical users, their characteristics, the tasks they seek to achieve, and their organizational and computing environments. After that, during the literature review about usability and human-computer interaction, 17 usability evaluation methods were selected: eight inspection methods and nine user testing methods.

To choose the methods that would be compared in this study, 11 criteria were established: procedure and typical location of data collection, data type and source, development phase and status, usability factors measured by the method, cost of evaluation, reliability of measurements, technical support needed, and additional costs, like users' involvement, materials, training. Among the evaluation methods pre-selected, four inspection methods were chosen, according to these criteria:

- a standard for the ergonomics and usability of information technology products - the ISO 9241-10 Dialogue principles (ISO, 1996);
- an ergonomic design guideline - the Ergonomic Criteria (Bastien & Scapin, 1993);
- a checklist - the Usability Index Checklist for Websites (Keevil, 1998); and
- the heuristic evaluation, whose support tool - the Corporate Portal Heuristics (Dias, 2001) - was elaborated for this research, based on the context of use of the sample corporate portal and on several principles or heuristics published by usability authors.

To compare not only the quality of the chosen methods, but also the effect of the evaluators' experience and abilities, the usability inspection of the sample corporate portal

was divided into two phases. The first phase involved 27 non-expert people, without any experience in software development, web design or usability evaluations, but familiar with computers and web sites, while the second phase involved 13 software engineers and eight usability experts.

Method

At first, the non-expert evaluators were assigned, at random, to five different groups: one called Control group (CNTL), and four other groups, each corresponding to a different inspection method. The Control group did not follow any method or document, while the others received the following documents related to these inspection methods:

- ISO - ISO 9241-10 Dialogue principles (ISO, 1996);
- EC - Ergonomic Criteria (Bastien & Scapin, 1993);
- INDEX - Usability Index Checklist for Websites (Keevil, 1998);
- HEU - Corporate Portal Heuristics (Dias, 2001).

All evaluators were invited to read the method's document, and to identify usability problems on the sample portal's interface, initially by exploring it on their own and then, by trying to perform three typical tasks. The tasks provided to all evaluators had been identified by the researcher, as being representative of typical tasks, based on detailed analysis of the context of use of the sample portal. The participants were then given an evaluation guide elaborated for this study, and asked to inspect, using the respective methods' documents, all web pages they walked through to complete the typical tasks.

Finally, each evaluator reported the usability problems they encountered on a common evaluation form, correlating each problem to a usability principle described in the method's document they used. On the same form, the evaluators answered 17 demographic questions, and expressed their opinion about the examined corporate portal and the usability evaluation method used, answering 13 satisfaction statements (on a semantic differential scale, with two opposite adjectives along each statement).

The usability problems identified through use of the four methods were categorized using common metrics, so data could be compared across methods on dimensions like type of usability problems (general or recurring problems; problems of low, moderate or high severity; and problems associated to usability factors).

To compare the evaluation methods, the following criteria, gathered from previous comparative studies, like Jeffries et al. (1991) and Bastien, Scapin & Leulier (1996), were defined in advance: average time spent by evaluators; researcher time spent on evaluation planning, sessions and data analysis; number and type of identified usability problems; number of correct correlations between identified problems and method's principles; size of aggregates (usability problems found by different number of evaluators); evaluators' doubts and average opinion about the methods.

Based on the preliminary results of the first phase, in regard to the comparison criteria above, two methods were selected to proceed on the second phase: the Corporate Portal Heuristics (HEU), and the Usability Index Checklist for Websites (INDEX). The software engineers were allocated to two groups, each corresponding to one of the methods. The same occurred to the usability experts. All four groups were invited to try to

perform the same typical tasks and to follow the same guidance given to non-expert evaluators.

After reducing all usability problems, described by all evaluators, to a list of unique problems, the researcher asked the participants to assign a severity rating to each problem listed, according to the following scale:

- 0 - this is not a usability problem;
- 1 - low severity problem (cosmetic);
- 2 - moderate severity problem (obstacle);
- 3 - high severity problem (catastrophe).

Research Results and discussion

During the usability inspection of the sample corporate portal, 56 web pages were visited. The participants of this research reported and described 701 usability problems, which were reduced to 120 unique problems, after verbal logic analysis of all descriptions. The unique problems were associated to numerical codes categorized into seven usability factors, derived from usability heuristics defined by Nielsen (1994):

- Consistency;
- Context compatibility;
- Error prevention;
- Flexibility and efficiency of use;
- Aesthetics and simplicity;
- User control;
- Visibility, recognition and conduction.

Due to overlapping between usability factors, 14 usability problems were classified in more than one category. For this reason, the total of usability problems, in some aspects of this data analysis, may be 134, corresponding to 14 duplicated problems added to the previously mentioned total of 120 unique problems.

Most of the usability problems reported (68%) were considered of moderate severity, according to evaluators' answers. 'User control' and 'Error prevention' problems got the highest average severity ratings (1,93 and 1,69, respectively - in a scale from 0 to 3). This result is compatible with Nielsen's statement : error situations or actions that do not follow user's commands are critical for usability because they represent situations where the user is in trouble and potentially unable to use the system to achieve the desired goal (Nielsen, 1993). Therefore, web pages need to behave according to user expectations, responding to his commands and mouse clicks, and avoiding errors. Otherwise, users may give up using the corporate portal to search information or to complete specific tasks.

Based on evaluators' opinions, context compatibility, legibility, efficiency, effectiveness and attractiveness were the positive aspects of the sample corporate portal. On the other hand, the confusing design of this web portal was the satisfaction item that reached the highest percentage of negative scores (40% of the evaluators' answers), for being such a usability obstacle to the completion of the typical tasks (17% of the evaluators

were not able to complete all three tasks). Some web pages displayed poorly on low resolution screens, requiring horizontal scrolling, and there was lack of format consistency between some of them.

To compare the selected evaluation methods, the criteria previously mentioned were analyzed in detail. In order to easily identify the evaluation groups, the following acronyms will be used throughout this paper:

Table 1 - Evaluation groups.

Non-experts	
ISO 9241-10 group	Non_E_ISO
Control group	Non_E_Cntl
Ergonomic Criteria group	Non_E_EC
Heuristic Evaluation group	Non_E_Heu
Usability Index group	Non_E_Index
Software engineers	
Heuristic Evaluation group	SE_Heu
Usability Index group	SE_Index
Usability experts	
Heuristic Evaluation group	UE_Heu
Usability Index group	UE_Index

Average evaluation time

Comparing the time spent by all evaluation groups, it was noticed that non-experts took more time to complete this activity than software engineers or usability experts. The largest average time (4:27h) was spent by the EC non-expert group, while the shortest one (1:11h) was that of the usability expert group using the Heuristics method. Among the Index groups, software engineers were the fastest group (avg time = 1:54h).

The time spent by the researcher was 46% smaller with software engineers and usability experts, in comparison to non-expert groups, because it was unnecessary to be present during evaluation sessions, and to correct evaluation form fields during data analysis of expert groups results.

No usability expert or software engineer asked the researcher to solve any doubts about the evaluation process, and no filling errors were identified on their evaluation forms. Allied to their previous computer and/or usability knowledge, the evaluation guide, sent by e-mail to these groups, was considered clear enough for their understanding of the steps to be followed.

Number and type of identified usability problems

Table 2 shows the distribution of usability problems found by all groups. The predominant types of usability problems were: 'Error prevention', 'Visibility, recognition and conduction', and 'Flexibility and efficiency of use'. Software engineers (SE) and usability experts (UE) using the Index method were the groups that identified more of these types of

usability problems. The Heuristics method, on the other hand, helped evaluators detect 'User control' and 'Aesthetics and simplicity' problems. Only five 'Context compatibility' and 11 'Consistency' problems were identified.

Table 2 - Usability problems identified.

Usability Factor	Total	Non_E ISO	Non_E Cntl	Non_E EC	Non_E Heu	SE_ Heu	UE_ Heu	Non_E Index	SE_In dex	UE_In dex
Error prevention	29	4	6	1	7	10	12	10	17	11
Visibility, recognition and conduction	28	4	8	6	10	11	9	13	15	14
Flexibility and efficiency of use	28	3	5	6	8	12	10	11	13	14
Aesthetics and simplicity	20	3	7	6	8	8	10	5	4	4
User control	13	4	3	3	7	7	7	2	3	3
Consistency	11	2	2	2	2	5	6	5	6	4
Context compatibility	5	1	1	0	2	0	1	1	3	2
Total (with duplicates)	134	21	32	24	44	53	55	47	61	52
Total (unique problems)	120	18	27	19	35	48	48	43	56	47

The ISO standard and the Ergonomic Criteria were the evaluation methods that presented the worst performances in number of identified problems: for example, the Non_E_EC group was able to identify just one of the 29 'Error prevention' problems; the Non_E_ISO group, four.

Among the evaluation groups of the first phase (non-experts), the Non_E_Index group was able to detect the largest number of 'Visibility, recognition and conduction', 'Flexibility and efficiency of use', 'Error prevention' and 'Consistency' problems. The Non_E_Heu group was the best on identifying 'Aesthetics and simplicity' and 'User control' problems, and the second best, among non-experts, on 'Visibility, recognition and conduction', 'Flexibility and efficiency of use' and 'Error prevention' usability factors.

Taking into account the 'Error prevention' factor, totaling 29 usability problems, the best performances were achieved by the SE_Index group (17 problems), followed by the UE_Heu group (12 problems). The 'Context compatibility' factor, because of its small number of identified problems (just five), could not offer enough data to any significant conclusions about evaluation groups' contribution.

Considering now the 'Flexibility and efficiency of use' factor (with a total of 28 detected problems), the usability experts and software engineers groups using the Usability

Index Checklist for Websites, followed by the software engineers using the Heuristics method, were the groups that identified more problems (14, 13 and 12 problems, respectively). The 'Aesthetics and simplicity' and 'User control' problems were more evident to the groups that used the Heuristics method, regardless of evaluator's qualification, while the 'Visibility, recognition and conduction' problems were detected in larger number by the Index method.

In relation to recurring problems, the Index groups surpassed all others in identifying usability problems that were replicated in more than six web pages. The SE_Index group was also the one that detected more usability problems occurring in just one web page. This result confirms Jeffries et al. study (1991): guidelines (or checklists, like the Index method) are good tools to find recurring and general problems, and may be used by software engineers to identify these types of usability problems when usability experts are not available.

Correlations between identified problems and method's principles

In order to check the quality and clearness of each method's document, each evaluator was asked to relate the usability problems he/she found to the corresponding concept, or principle, described in the document he/she employed (ISO 9241-10 standard, EC guidelines or Corporate Portal Heuristics). The Index method was not included in this comparison because all problem descriptions of this method are directly associated to a principle in its spreadsheet. Table 3 shows that the highest correlation score was achieved by usability experts (UE) - 96 % (90 correct associations in a total of 94 described problems). This reveals better skills of usability experts in using the Heuristics method, when compared to software engineers or non-experts. On the other hand, among the 19 problems described by the ISO group, only five were correctly associated to the standard's principles, due to misunderstanding of its concepts.

Table 3 - Correlation between problems and principles.

	Non_E_ISO	Non_E_EC	Non_E_Heu	SE_Heu	UE_Heu
Correct relation	5 (26%)	23 (68%)	37 (73%)	58 (73%)	90 (96%)
Incorrect or no relation	14 (74%)	11 (32%)	14 (27%)	22 (27%)	4 (4%)
Total problem descriptions	19 (100%)	34 (100%)	51 (100%)	80 (100%)	94 (100%)

Size of aggregates

Figure 1 compares the evaluation methods, relating the number of evaluators to the number of detected problems. Among all evaluators, experts or non-experts, the SE_Index group was the one that identified the largest number of usability problems, achieving, with only three evaluators, 45,8% of the total of 120 unique problems detected by all evaluators (55 problems). Any other group did not accomplish similar results, even with six or seven people.

The worst performances, in number of problems found, were attained by the non-expert groups EC and ISO, whose curves present a similar behavior, lower than the Control group's curve. This finding differs from Bastien, Scapin & Leulier (1996), whose study verified a better performance of the EC guideline over the ISO standard, and also over the control group, when applied to musical databases.

Analyzing just the non-expert curves, Figure 1 shows that the Non_E_Index group identified the largest number of usability problems (43 problems), equal to 57,3% of the total of 75 unique problems detected by all non-expert groups. Just one evaluator of this group was able to find more problems than all evaluators of Non_E_ISO or EC groups, taken separately.

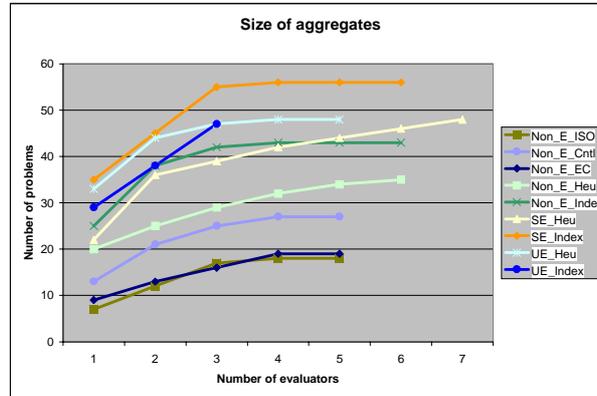


Figure 1 – Size of aggregates.

Examining the behavior of all non-expert curves, it is noticed that, except the Non_E_Heu group, there were no additional problems detected by more than four evaluators (curves parallel to X-axis). The Non_E_Heu group's curve presents a smooth slope, due to the gradual contribution of each additional evaluator and tends to stabilize in 35 problems, after the sixth evaluator. This figure also shows that, to get 40% of the 75 unique problems identified by all non-experts (30 usability problems), it was necessary to assign two members of the Non_E_Index group or three of the Non_E_Heu group, while the other groups did not attain this level, even with five evaluators.

Analyzing the behavior of all curves, one may notice that there is a certain similarity between groups using the same method, in relation to slope and stability points. Taking the Index groups, for example, the SE and UE curves have the same slope until the third evaluator, and the Non_E and SE curves stabilize also after the third evaluator. Because of the small number of people in the UE_Index group, it was not possible to define its stability point.

Similarly, the Non_E_Heu and SE_Heu curves suggest a certain parallelism between them right after the second evaluator, and present a smoother slope due to the gradual contribution of each additional evaluator. Among the groups that used the Heuristics method, the UE group was the one that found more usability problems,

reaching, after the third evaluator, 47 problems (39,2% of the total problems detected). To reach this level, it was necessary to assign seven software engineers. The Non_E_Heu curve implies that, even adding more evaluators, the number of problems detected by the UE_Heu group would not be equalled.

Comparing the UE curves until the third evaluator, it was verified that both groups reached the same number of problems at this point. The software engineers, on the other hand, found a larger number of usability problems when using the Index method.

Average severity of usability problems identified by each method

Table 4 shows the average severity, in a scale from 0 to 3, of the usability problems found by each group. There was a convergence to moderate severity. The Index groups, however, reached a lower average severity, around 1,4. This finding is in accordance with Jeffries et al. (1991), whose study points, as a disadvantage of inspections based on guidelines (or checklists), the fact that they miss many high severity problems, when compared to heuristic evaluation, for example. Among the Index groups, the software engineers found a larger number of high severity problems (Table 5).

Table 4 – Average severity.

	Non_E _ISO	Non_E _Cntl	Non_E _EC	Non_E _Heu	SE_ Heu	UE_ Heu	Non_E _Index	SE_Index	UE_Index
Average severity	1,8	1,72	1,75	1,74	1,6	1,72	1,4	1,4	1,37

Table 5 – Identification of high severity problems.

Total of 18 high severity problems	Non_E _ISO	Non_E _Cntl	Non_E _EC	Non_E _Heu	SE_ Heu	UE_ Heu	Non_E _Index	SE_Index	UE_Index
Number of high severity problems	4	6	4	9	7	10	5	7	6

Considering the 18 high severity problems (Table 5), rated by all evaluators, the UE_Heu group was the one that detected the largest number, surpassing the Index group of software engineers, that found the largest number of problems in total (Table 2).

Comparing only non-expert evaluators, the Heuristics group was also the one that reached the largest number of problems in this range (50% of the high severity problems detected by all groups), while the ISO and EC groups identified only four. One evaluator of the Non_E_Heu group found 5 severe problems, a performance comparable or superior to the total number of severe problems detected by Non_E_ISO, EC and Index groups (4, 4 and 5, respectively).

Taking into account the behavior of the aggregates' curves for high severity problems, this study also observed that there was a certain resemblance between groups using the same method in relation to slope and stability points, similarly to what happened with the total of usability problems, shown in Figure 1.

Evaluators' average opinion about the methods

After the evaluation process itself, the participants of this study were asked to give their subjective opinion about the methods they used. Comparing their answers to questions associated to a semantic differential scale, it was verified that the Corporate Portal Heuristics presented more positive and less negative aspects than the other methods, while the Ergonomic Criteria got just the opposite.

The EC method received the most negative evaluation, being considered boring, difficult, confusing and excessively detailed. The only question that this method got exclusively positive ratings was about its text, considered well written by all evaluators. One of the evaluators mentioned that this method should be less detailed and less redundant in order to be well understood by readers.

The evaluators that used the Corporate Portal Heuristics, on the other hand, considered the following positive aspects of this method: utility, effectiveness, context compatibility, well written text, easiness, efficiency and flexibility. One evaluator mentioned that this method is simple and easy to understand.

Based on the opinion of ISO group members, this method was referred as efficient, effective, and with a well written text. However, some answers tended to negative adjectives, like rigid, horrible and frustrating. One of the evaluators of this group mentioned that this method was so complicated that he/she preferred to take part in the control group and evaluated the sample portal following his/her own judgements.

The positive aspects of the Usability Index Checklist for Websites were: well written text, efficiency, utility, effectiveness and context compatibility. However, around 40% of the evaluators considered it boring and excessively detailed. One Index evaluator remarked that many items of this checklist were too general for this specific usability inspection. Another evaluator commented that this method is a little bit confusing and inflexible.

Conclusion

Considering the results of the comparative analysis, two evaluation methods were remarkable in this case study: the Heuristic Evaluation (whose support tool was the document entitled Corporate Portal Heuristics), and the Usability Index Checklist for Websites, mainly for being able to identify more problems, and to cover all usability factors and severities, when applied as complementary methods. The INDEX groups identified more general, 'Error prevention', 'Flexibility and efficiency of use', 'Visibility, recognition and conduction' problems, while the HEU groups were the best in identifying severe, 'User control' and 'Aesthetics and simplicity' problems.

It was noticed that usability experts, using the Corporate Portal Heuristics, and software engineers, using the Usability Index Checklist for Websites, were the evaluation groups that identified more usability problems, and in less time, when compared to non-expert groups. The time spent by the researcher during the evaluation sessions and data analysis was also smaller with usability experts and software engineers.

Taking into account the quality of each method's document, and the evaluators' subjective opinion about them, the Corporate Portal Heuristics got more positive results than the other tested methods, and the usability expert group was the best one in

corresponding the detected problems to the correct principles presented in the methods' documents.

In regard to the size of aggregates of non-expert groups, this study verified that the EC and the ISO groups, even with five evaluators each, did not achieve the same number of usability problems uncovered, individually, by the first evaluators of the HEU or INDEX groups. The performances of the EC and ISO groups were compared to each other, but below that of the Control group. This particular result is at odds with Bastien, Scapin & Leulier (1996) data. These researchers observed that, when using the EC method, participants uncovered more usability problems than ISO or Control groups.

In this study, it was also observed that there was no relevant increase on the total number of uncovered problems, as more than three or four evaluators were added to the group. The same happened with the aggregates' curve of severe problems. It was also verified that usability experts got better performance with fewer people, when compared to software engineers or non-experts.

The findings of this research were confronted with other comparative studies, like Bastien, Scapin & Leulier (1996), who compared the Ergonomic Criteria with the ISO 9241-10 Dialogue principles; Jeffries et al. (1991), who evaluated four techniques (among them, heuristic evaluation and software guidelines); Karat, Campbell & Fiegel (1992), who investigated the relative effectiveness of empirical usability testing and individual and team walkthrough methods in identifying usability problems; and Levi & Conrad (1996), who examined the heuristic evaluation using two different groups of evaluators - software developers and usability experts. In general, this case study remarked that:

- (1) the heuristic evaluation achieved better results, when compared to guidelines, identifying a larger number of severe problems, specially when performed by usability experts (agreeing with Jeffries et al. (1991));
- (2) assigning more than three or four evaluators does not necessarily, or significantly, increase the total number of usability problems identified. In heuristic evaluations, it would be better to assign usability experts as evaluators, in order to obtain a better performance with fewer people (compatible with Nielsen's suggestion to employ from three to five usability experts in heuristic evaluations (Nielsen, 1993));
- (3) software engineers, following guidelines or checklists, can complete usability evaluations with relative success, uncovering general and recurring problems, but missing some severe problems (confirming the results of Jeffries et al. (1991); and Karat, Campbell & Fiegel (1992));
- (4) one method can not completely substitute another one, because they identify different types of problems. Usability evaluation methods should be taken as complementary methods (agreeing with Karat, Campbell & Fiegel (1992));
- (5) each professional profile is more suitable to a certain method: software engineers, to checklists; usability experts, to heuristic evaluation (diverging from Levi & Conrad (1996) data - these researchers did not find significant differences between these two groups of evaluators).

After comparing the examined methods, this research made clear that, to achieve satisfactory results in the usability evaluation of this case study's corporate portal, identifying more problems, in quantity and in variety, taking less time and fewer evaluators, it was necessary to analyze its context of use and to employ, as complementary

methods, the Usability Index Checklist for Websites, used by software engineers; and the heuristic evaluation, performed by usability experts, using the Corporate Portal Heuristics.

The results of this research suggest that, in usability evaluations of corporate portals, it would be more cost-effective to assign, at first, software engineers, using the Index checklist. After proper modification of the portal design, eliminating the usability problems identified in the first step, some usability experts would be assigned to perform heuristic evaluation and complement the usability inspection. Deciding between these methods or procedures, however, depends on the evaluation goals, the portal's context of use, and the available resources.

References

- Bastien, C. & Scapin, D. (1993) Critérios ergonômicos para avaliação de interfaces homem-computador. [on-line], September, 2000. <http://www.labiutil.inf.ufsc.br/indice-1.html>.
- Bastien, C., Scapin, D. & Leulier, C. (1996) Looking for usability problems with the ergonomic criteria and with the ISO 9241-10 dialogue principles. In: *Proceedings of CHI'96, Vancouver, Canada, April, 1996*. [on-line], September, 2000. http://www.acm.org/sigchi/chi96/proceedings/intpost/Bastien/jmcb_txt.htm.
- Dias, C. (2001) *Métodos de avaliação de usabilidade no contexto de portais corporativos: um estudo de caso no Senado Federal*. Brasília: UnB. 229p.
- ISO (1996) 9241 Part 10. *Ergonomic requirements for office work with visual display terminals, Part 10: Dialogue principles*.
- Jeffries, R. et al. (1991) User interface evaluation in the real world: a comparison of four techniques. In: *Proceedings of CHI'91, New Orleans, LA, April-May, 1991*. New York: ACM. p. 119-124.
- Karat, C., Campbell, R. & Fiegel, T. (1992) Comparison of empirical testing and walkthrough methods in user interface evaluation. In: *Proceedings of CHI'92, Monterey, CA, May, 1992*. New York: ACM Press. p. 397-404.
- Keevil, B. (1998) Measuring the usability index of your web site. In: *Proceedings of CHI'98, Los Angeles, CA, April, 1998*. New York: ACM Press. p. 271-277.
- Levi, M. D. & Conrad, F. G. (1996) A heuristic evaluation of a World Wide Web prototype. *Interactions*, July/Aug., 1996. p. 50-61.
- Nielsen, J. (1993) *Usability engineering*. Boston, MA: Academic Press. 362p.
- _____. (1994) Ten usability heuristics. In: Nielsen, J. & Mack, R. (eds.) *Usability inspection methods*. New York: John Wiley & Sons. [on-line], June 1999. http://www.useit.com/papers/heuristic/heuristic_list.html.

Design de ambientes para EaD: (re)significações do usuário

Heloísa Vieira da Rocha¹, Janne Yukiko Y. Oeiras¹,
Fernanda Maria Pereira Freire², Luciana Alvim Santos Romani³

¹Instituto de Computação - Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6176 - CEP 13083-970 - Campinas - SP - Brasil

²Núcleo de Informática Aplicada à Educação (NIED) - UNICAMP
Cidade Universitária "Zeferino Vaz" - Bloco V da Reitoria - 2o. Piso
CEP13083-970 - Campinas - SP - Brasil

³Embrapa Informática Agropecuária
Caixa postal 6041 - CEP 13083-970 - Campinas - SP - Brazil

{heloisa,janne}@ic.unicamp.br, ffreire@nied.unicamp.br,
luciana@cnptia.embrapa.br

Abstract. *This paper describes and discusses data related to the alternative use of some tools of TelEduc's distance education environment. This data was collected during several distance courses and its analysis indicates that the users re-interpret designer's intentions and develop a different model if compared to the conceptual model of environment's design. Contrary to the IHC results, this possibility of several re-interpretations by the user is desirable, considering the process of learning and teaching at distance.*

Resumo. *Este trabalho descreve e discute alguns dados relacionados ao uso alternativo de algumas ferramentas do ambiente TelEduc para EaD, coletados no decorrer de vários cursos a distância. A análise dos dados indica que os usuários reinterpretem as intenções do designer desenvolvendo um modelo muitas vezes distante do modelo conceitual do design do ambiente. E, de certa forma contraditoriamente aos resultados da área de IHC, esta possibilidade de múltiplas reinterpretações por parte do usuário é desejável, considerando-se a tarefa de educar a distância.*

1. Introdução

Com o avanço tecnológico e a consolidação da Internet como meio eficiente de comunicação, pesquisadores no mundo todo vislumbram na rede uma oportunidade ímpar de suporte a inovações no processo educacional. O trabalho de pesquisa de vários educadores e cientistas da computação resultou na possibilidade de várias pessoas acessarem salas de aula virtuais, grupos de trabalho na rede, campi eletrônicos e bibliotecas *online* num espaço compartilhado.

Os últimos anos de pesquisa foram marcados pelo surgimento de inúmeras ferramentas computacionais dirigidas a EaD em todo o mundo. Algumas obtiveram mais sucesso e passaram a ser exploradas comercialmente, outras são de uso restrito das instituições que as desenvolveram. Dentre elas, tornaram-se mais populares os ambientes

para autoria e gerenciamento de cursos a distância na Internet, como por exemplo o WebCT [Goldberg et al., 1996] e o Lotus Learning Space [Lotus, 2001]. Esses ambientes objetivam facilitar o processo de oferecer cursos pela rede possibilitando que um formador não precise se tornar um especialista em computação ou tecnologia Web para elaborar e disponibilizar material didático, bem como para acompanhar o desenvolvimento de seus alunos. Esses ambientes são formados pela junção de várias tecnologias de comunicação mediadas por computador (CMC), tais como o correio eletrônico e os sistemas de conferência por computador, aliados a outros recursos da Web.

De forma geral, as ferramentas que compõem esses ambientes estão organizadas de acordo com suas funcionalidades e controles de acesso em: autoria, administração e uso dos alunos. No conjunto de autoria há um número grande de ferramentas para edição e inclusão de textos, slides ou transparências, áudio, vídeo e animações. Além disso, elas também possibilitam ao professor definir cores, padrão das páginas e quais recursos de comunicação poderão ser usados durante o curso. O grupo referente a administração inclui ferramentas que facilitam o gerenciamento do curso e fornecem informações ao formador a respeito do seu andamento. Esses dois grupos estão disponíveis apenas para o formador e seus auxiliares. O conjunto de recursos disponíveis para os alunos inclui ferramentas para comunicação, avaliação automática, pesquisa em glossários, anotações, criação de páginas pessoais e acompanhamento de resultados de avaliações.

Como todo ambiente computacional com fins educacionais, estes também seguem uma metodologia e uma concepção a respeito do que se entende por aprendizagem. Assim, existem ambientes mais abertos e flexíveis e outros que impõem tanto ao professor como ao aluno uma seqüência restrita de ações. Desta forma, pode-se encontrar ambientes que mapeam diretamente a metodologia usada na sala de aula tradicional presencial para a sala virtual; aqueles baseados em resolução de problemas; os que se apresentam no formato de tutoriais, entre outros.

As ferramentas dos ambientes abertos, a exemplo do TelEduc, são desenvolvidas com um certo propósito e intenção por parte dos desenvolvedores e pesquisadores que se ocupam da área de formação de profissionais, o que vem a se constituir seu modelo conceitual [Rocha e Baranauskas, 2000]. No entanto, suas ferramentas podem ser exploradas e utilizadas de diferentes formas, a depender dos objetivos, abordagem pedagógica adotada e perfil dos usuários. À essa forma particular de utilizar uma determinada ferramenta denominamos *(re)significação*. Em outras palavras, cada ferramenta é concebida com uma determinada funcionalidade dentro de uma visão específica do que vem a ser a tarefa de educar, mas o modo de utilizá-la em um dado contexto pode gerar, como será discutido neste artigo, outras funções de acordo com a significação a ela atribuída pelo usuário.

A seguir são apresentadas as ferramentas do ambiente TelEduc e suas respectivas funcionalidades de acordo com o modelo conceitual de design. Na seção 3 são descritos e analisados dados que sugerem novas funcionalidades para algumas ferramentas do ambiente. Finalmente, na seção 4, apresentamos algumas considerações a respeito da importância de se analisar os múltiplos modelos do usuário para o desenvolvimento de ambientes de EaD.

2. O Ambiente TelEduc

O TelEduc é um ambiente para EaD que foi desenvolvido tendo como alvo a formação de professores para a Informática na Educação. Seu desenvolvimento teve início em 1997 no Núcleo de Informática Aplicada à Educação¹ (NIED) em parceria com o Instituto de Computação² (IC), ambos da UNICAMP, apoiando-se em estudos e pesquisas a respeito do processo *construcionista* de formação de professores, que enfatiza entre outros aspectos, o *conhecimento contextualizado* [Freire e Prado, 1996; Valente, 1998, Valente, 1999]. Assim, é fundamental que o professor possa durante a formação manter-se em sua escola: ponto de partida para sua motivação, reflexão e aprendizagem. No entanto, esta forma de trabalho quase sempre gera problemas operacionais importantes considerando-se a inviabilidade de se deslocar o formador de uma escola para outra. Com o objetivo de resolver esse impasse deu-se início ao Projeto TelEduc visando o desenvolvimento de ferramentas que propiciassem a formação desses profissionais a distância.

Todo o desenvolvimento do ambiente foi participativo, ou seja, a construção do modelo conceitual, a partir da análise da tarefa, foi feita conjuntamente com professores formadores do NIED. Analisar a tarefa de educar é um processo altamente complexo considerando-se as especificidades que envolvem o entendimento dessa tarefa quando da mudança de pressupostos pedagógicos e/ou áreas de conhecimento. Partiu-se então analisando a situação presencial e levantando os requisitos de interação necessários para que se pudesse a distância, por em prática a metodologia construcionista usada pelos formadores do NIED. Com isso tínhamos claro que a interação constante entre os participantes de um curso era fundamental, e que de modo algum poderíamos ter um ambiente rígido, no sentido de dirigir a ação dos formadores e/ou alunos. Basicamente, estávamos descartando a EaD estilo “pombo correio” e construindo um espaço de trabalho que possibilitasse o “estar junto virtual” [Valente, 1999]. A partir dessa fase de análise, iniciou-se a implementação incremental do ambiente cuja forma atual passamos a descrever sucintamente a seguir.

A página de entrada do TelEduc é dividida em duas partes. À esquerda estão as ferramentas que serão utilizadas durante o curso e, à direita é apresentado o conteúdo correspondente à ferramenta selecionada na parte esquerda. Ao se entrar no curso, é apresentado o conteúdo da ferramenta **Agenda** que contém orientações do que deve ser feito em um determinado período, informações atualizadas, dicas ou sugestões dos formadores para os alunos. Essa página funciona como um canal de comunicação (diária, semanal, mensal, etc.) direto entre formador e aluno (figura 1).

Em diferentes momentos de um curso, o formador pode utilizar um subconjunto qualquer de ferramentas, que fica visível no menu à esquerda. O oferecimento ou não de cada uma delas faz parte da metodologia adotada pelo formador.

O conjunto total de funcionalidades oferecidas pelo TelEduc pode ser dividido em três grandes grupos: *ferramentas de coordenação*, *ferramentas de comunicação* e *ferramentas de administração*.

¹ <http://www.nied.unicamp.br>

² <http://www.ic.unicamp.br>

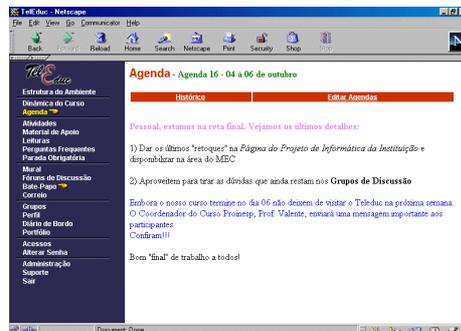


Figura 1: Página de entrada do TelEduc

As *ferramentas de coordenação* foram concebidas para organizar e subsidiar as ações de um curso. O conjunto engloba as ferramentas **Agenda**, descrita anteriormente; **Histórico**, que armazena de forma seqüencial todas as agendas de um curso e **Dinâmica**, na qual o formador descreve a proposta do curso indicando o tempo de duração, os objetivos, a forma de avaliação, etc. Desse grupo também fazem parte as ferramentas **Leituras**, **Material de Apoio**, e **Atividades**. A interface de **Leituras** e **Material de Apoio** é semelhante, mas elas diferem conceitualmente: a primeira é usada para disponibilizar textos e material bibliográfico do curso, enquanto a segunda, pode ser usada para disponibilizar qualquer material que auxilie na resolução de uma determinada atividade.

A **Parada Obrigatória** tem uma forte vinculação com a abordagem pedagógica usada pelo NIED em seus cursos e funcionalmente é análoga à ferramenta **Atividades**. Conceitualmente, seu uso se dá em momentos do curso em que o formador tem necessidade de fazer um resgate dos principais pontos até então tratados. Trata-se de uma atividade especial que procura explorar todo o conteúdo já visto até um determinado momento do curso, integrando atividades e leituras que o aluno pode ter percebido não relacionadas. Finalmente, nesse grupo, encontramos ainda a ferramenta **Perguntas Frequentes** para que o formador vá organizando as dúvidas gerais que aparecem no decorrer do curso e a ferramenta **Grupos** que possibilita organizar os alunos em subgrupos de trabalho, quando for conveniente.

No conjunto de *ferramentas de comunicação* temos o **Correio Eletrônico**, o **Bate-papo** e **Fóruns de Discussão**, implementadas no mesmo formato que as usadas comumente na Internet. Todas são internas ao ambiente, ou seja, para se ter acesso às mensagens do **Correio** é preciso estar conectado ao TelEduc. Os formadores tem total liberdade de criar e eliminar os **Fóruns de Discussão**, de acordo com tópicos que julgue relevantes serem discutidos mediante este tipo de ferramenta. As sessões de **Bate-papo** são agendadas previamente e registradas: qualquer participante do curso pode ter acesso aos registros para posterior análise da discussão. Além dessas, temos a ferramenta denominada **Mural** que possibilita que recados gerais - comumente colocados em murais reais - como aviso de eventos, links interessantes encontrados na Internet, etc., possam ser anexados por qualquer participante do curso.

Também como *ferramenta de comunicação* temos o **Portfólio** que computacionalmente pode ser visto como um espaço em disco para o aluno no servidor TelEduc. Seu uso é prover um mecanismo para o aluno comunicar ao grupo ou ao formador o resultado de seu trabalho e receber comentários e sugestões. Neste espaço o aluno disponibiliza textos, respostas de atividades, URL, etc. e decide que tipo de compartilhamento deseja: o *totalmente compartilhado* possibilita que todos os participantes do curso possam ter acesso e comentar seu conteúdo; analogamente o modo *compartilhado com formadores* irá permitir acesso somente ao grupo de formadores do curso e o *não compartilhado* veta qualquer tipo de acesso. Geralmente, essa última opção é usada quando o aluno ainda não atingiu o resultado que espera, ou seja, é um trabalho em andamento que está ocupando espaço de armazenamento. Duas outras ferramentas que podem ser consideradas de comunicação mas que são bastante específicas da metodologia de formação em que se baseia o TelEduc são o **Diário de Bordo** e o **Perfil**. A primeira, objetiva que o aluno use esse espaço para refletir sobre seu processo de aprendizagem. Daí o nome sugerir um diário, como os de bordo que indicam o percurso seguido ao longo do curso. A ferramenta **Perfil**, por sua vez, é usada para o aluno se apresentar ao grupo de forma bastante pessoal, colocando sua foto, dizendo quem é, do que gosta, o que faz, seus hobbies, sua família, sua cidade, etc. [Romani et al., 2000].

Finalmente, temos o terceiro grupo que é o das *ferramentas de administração* em que englobamos as ferramentas de apoio ao formador no gerenciamento da parte administrativa do curso – alunos, inscrições, datas de inscrição, formadores etc., ferramentas de apoio à autoria, ou seja, as que permitem ao formador transferir para o TelEduc todo o material didático que necessita, além de atualizar a **Agenda**, a **Dinâmica** e possibilitar a seleção de ferramentas, etc.. Também nesse grupo encontram-se ferramentas que auxiliam o formador a verificar os **Acessos** dos alunos ao ambiente em seus diferentes componentes. Essa ferramenta tornou-se necessária porque não se conseguia identificar o "aluno calado e presente" do "aluno ausente", sendo esta diferenciação extremamente importante para o acompanhamento do curso. Desta primeira necessidade surgiu o principal tópico atual de interesse de pesquisa e desenvolvimento no TelEduc: *mostrar as pessoas*. Nessa linha de pesquisa, já dispomos de uma primeira ferramenta, a **InterMap**, que já está implementada [Romani, 2000].

3. Modelo do usuário: exemplos de (re) significação

Apresentamos a seguir dados relativos a dois grupos de ferramentas do TelEduc. No primeiro deles (3.1) é mostrada a forma idiossincrática como alguns usuários utilizam ferramentas de comunicação a distância do tipo **Bate-papo** e **Correio eletrônico**, ambas de ampla utilização na Internet e, portanto, familiares a eles. O segundo grupo (3.2) descreve exemplos de utilização das ferramentas **Parada Obrigatória** e **Diário de bordo**, especialmente desenvolvidas para apoiar ações educacionais a distância no ambiente TelEduc, cuja concepção está fortemente embasada, como já foi dito, na metodologia de formação de profissionais desenvolvida pela equipe do NIED da UNICAMP.

3.1. Bate-papo e correio eletrônico: usos (in)comuns

Os usuários de Internet geralmente conhecem a ferramenta de **Bate-papo** mesmo quando não são assíduos frequentadores desse tipo de ambiente virtual. Quase sempre as salas de **Bate-papo** se destinam a encontros sociais: as pessoas conversam informalmente sobre

vários assuntos, acabam se conhecendo e, dependendo dos interesses, passam a se "encontrar" com certa periodicidade. Essa forma de comunicação síncrona acaba gerando grupos de usuários que se transformam em "freqüentadores" de uma ou outra sala, bem aos moldes dos encontros sociais presenciais (barzinhos, clubes, turma da escola, etc.).

No caso do TelEduc, a inclusão da ferramenta de **Bate-papo**, surgiu como uma possibilidade de se discutir *online* tópicos de interesse de um determinado curso, havendo um agendamento prévio da sessão e a indicação de um assunto a ela relacionado. Exemplo dessa forma de utilização foi o que observamos durante o oferecimento de um curso cuja proposta de trabalho previa a composição de pequenos grupos. O **Bate-papo**, neste caso, foi usado para o grupo *tomar decisões*. Nem todos os participantes no entanto foram favoráveis à eleição dessa ferramenta como mecanismo de discussão, como pode ser observado no depoimento a seguir³:

```
Já falei algumas vezes, e pode parecer chatice, mas continuo
encanada com essa obrigatoriedade de chat síncrono para
decisões....

Sinto-me literalmente perdida com essa obrigatoriedade... não tenho
como estar presente nos horários propostos e, mesmo a grade tendo
sido feita, os horários não variam, p/ dar "chance" a todos.

Isso realmente está me incomodando...
```

A opinião dessa participante⁴ encontra respaldo na literatura que aponta como uma das vantagens do ensino a distância baseado na Web, a possibilidade de cada participante acessar seu curso a qualquer hora do dia, nos sete dias da semana [Harasim et al., 1996]. No entanto, cremos que o aparente consenso a respeito do agendamento de um *chat* nestas circunstâncias específicas está relacionado ao estilo daqueles que o marcaram, profissionais cujo trabalho depende fundamentalmente de recursos da Informática e que, por esta razão, trabalham o dia todo frente ao computador e têm acesso irrestrito à rede, características que não se aplicam à participante do depoimento acima. Esse dado é importante à medida que revela que a eleição e funcionalidade de uma ferramenta estão fortemente relacionadas ao perfil do usuário, que pode estar ou não em conformidade com o seu design original.

Há outros dois fatores ainda que podem ter contribuído para que o **Bate-papo** fosse eleito como a ferramenta de comunicação preferencial por este grupo. O primeiro deles se deve a uma mudança do modelo de curso a distância, voltado para o trabalho colaborativo via rede. Os cursos a distância anteriores ao surgimento da Internet - via correio convencional, difusão por rádio e televisão, por exemplo - eram estritamente centrados na relação professor - aluno e não se dispunha de recursos tecnológicos que permitissem o agrupamento dos participantes. Outro aspecto que aqui importa é que a tomada de decisão requer a argumentação de cada participante que, quase sempre, vai se constituindo no decorrer da conversação. Trata-se portanto, de uma interação que se dá "*de forma natural e informal, tendo em vista que é relativamente não-planejada, ou seja, a construção da interação vai sendo 'planejada e re-planejada a cada novo lance do jogo da linguagem'*" [Dionísio, 2001, p. 74], diferentemente do que, provavelmente, aconteceria se a opinião do

³ A participante revelou sua insatisfação com o grupo em várias ferramentas: no Correio, no Portfólio e no Diário de bordo, do qual extraímos o trecho acima.

⁴ Possivelmente esta era também a posição de outras pessoas que não participaram da sessão marcada. Talvez elas não tenham se manifestado de forma explícita pelo fato de não estarem preocupadas em *tomar as decisões*, contentando-se em *saber as decisões tomadas* por meio do registro da sessão.

participante tivesse que ser postada no **Correio eletrônico** ou disponibilizada no **Portfólio**. O agendamento desse **Bate-papo** se assemelha à convocação de uma assembléia: *uma instância de discussão que tem o poder de deliberar a respeito de assuntos importantes*. Isso aponta para a necessidade de se ter design alternativos de comunicação síncrona que possibilitem mecanismos de coordenação podendo-se ter assembléias, seminários, etc.

Observa-se também que o **Bate-papo** do TelEduc é utilizado de forma *descompromissada*, similar ao que ocorre em outras salas da Internet em geral. Provavelmente isto se deve à *competência pragmática* dos usuários de CMC [Maingueneau, 1998]. Este tipo de conhecimento permite às pessoas saber como se portar no contexto particular de um **Bate-papo**, por exemplo⁵. Evidência disso é a frequência com a qual vários usuários, durante diferentes cursos, *entram e saem* da sala de **Bate-papo** à procura de outros participantes, como mostra a figura 2.

The image shows two overlapping windows from the TelEduc interface. The primary window, titled 'Bate-papo', has a section for 'Ver sessões realizadas - Seleção de Período' with input fields for 'Início' (16/04/2001) and 'Término' (30/05/2001), and buttons for 'Exibir' and 'Voltar'. Below this is a table of sessions:

Assunto	Início	Fim
Sessão de 16/04/2001 09:21hs até 16/04/2001 09:21hs	16/04/2001 - 09:21	16/04/2001 - 09:21
Sessão de 16/04/2001 10:00hs até 16/04/2001 10:02hs	16/04/2001 - 10:00	16/04/2001 - 10:02
Sessão de 16/04/2001 20:37hs até 16/04/2001 20:38hs	16/04/2001 - 20:37	16/04/2001 - 20:38
Sessão de 17/04/2001 11:26hs até 17/04/2001 11:26hs	17/04/2001 - 11:26	17/04/2001 - 11:26
Sessão de 17/04/2001 14:55hs até 17/04/2001 14:55hs	17/04/2001 - 14:55	17/04/2001 - 14:55
Sessão de 17/04/2001 14:55hs até 17/04/2001 14:55hs	17/04/2001 - 14:55	17/04/2001 - 14:55

The secondary window, also titled 'Bate-papo', shows a chat log with the following messages:

- 09:21:00 FabioFF entra no batepapo...
- 09:21:00 FabioFF sai do batepapo...
- 10:00:00 FabioFF fala para Todos: Bom Dia!!! Alguem mais no sala?
- 10:00:00 FabioFF fala para Todos: Ok?
- 10:02:00 FabioFF fala para Todos: Vou sendo...
- 10:02:00 FabioFF sai do batepapo...

Figura 2: Entra e sai do Bate-papo

Não se pode deixar de considerar, igualmente, a influência da interface do ambiente sobre o *entra e sai* no **Bate-papo**. A maior parte das ferramentas do TelEduc é ativada imediatamente quando o usuário clica sobre o nome de uma delas. As ferramentas são então sucessivamente substituídas. O **Bate-papo** funciona de forma diversa: ao entrar na sala abre-se uma nova janela que pode ser ou não minimizada. Pode-se supor que o usuário é induzido a sair do **Bate-papo**, uma vez que nos demais casos, ativar uma ferramenta implica necessariamente na desativação imediata da anterior.

Por outro lado, esse dado é um indicativo importante da falta de uma ferramenta no ambiente que informe ao usuário a *co-presença* de seus colegas de curso [Preece, 2000]. Essa conclusão é respaldada por outro dado que mostra o uso combinado da ferramenta **Acessos** e **Correio eletrônico**. A ferramenta **Acessos**, como explicado, permite obter informações a respeito da data e hora do último acesso de cada usuário. Observamos durante um de nossos cursos que um participante obtinha essa informação como um meio alternativo de saber se haviam outras pessoas conectadas ao ambiente como mostra a figura 3.

⁵ No contexto de EaD entendemos que fazem parte da competência pragmática dos usuários de CMC "established rules of behavior, conduct, and expression" que podem auxiliá-los a "know how to behave in the online space, and how to expect others to behave" (Haythornthwaite et al., 2000, p. 4).

Horário	Participante
15:05:52	Rômulo Maia Alves
15:27:08	Luciana Alvim Santos Romani
15:35:33	Fernanda Maria Pereira Freire
15:35:48	Mávia Beatriz Rodrigues Freire-da Cunha

Figura 3: Ferramenta Acesso mostrando o horário de utilização do ambiente de dois participantes

Entretanto essa informação não é suficiente para fazer contato com o outro usuário. Neste caso, o participante envia uma mensagem via **Correio eletrônico** como forma de "chamar o outro", como mostra a figura 4.

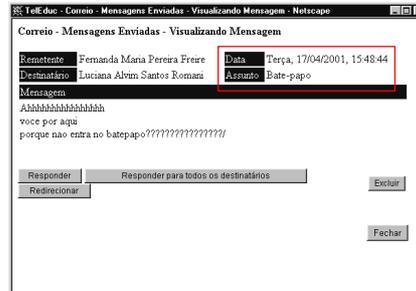


Figura 4: Mensagem no Correio convocando a participação do outro

Esse dado torna evidente dois pontos importantes: a (re)significação da ferramenta **Acessos** que em princípio nada tem a ver com a comunicação entre as pessoas no ambiente e a necessidade que os usuários têm em manter relações de proximidade mesmo em ambientes a distância, chegando a fazer uso quase-síncrono de uma ferramenta de comunicação assíncrona como o **Correio eletrônico**⁶. Fatos similares a este último são relatados por Haythornthwaite et al., (2000) a respeito da formação de comunidades de aprendizagem em cursos a distância.

3.2. Parada Obrigatória e Diário de Bordo: nomes como pontos de referência

Em relação ao conjunto de ferramentas desenvolvido para dar suporte a atividades de ensino-aprendizagem no TelEduc vamos destacar o uso inusitado que fez um formador da **Parada Obrigatória**. Como já descrito, essa ferramenta surgiu para atender uma necessidade metodológica: como um meio de dar destaque às principais idéias, conceitos ou temas relevantes num dado momento do processo de ensino-aprendizagem. O nome da ferramenta tem o propósito de chamar a atenção do formando para a importância de refletir sobre o material que nela estiver disponível, ou seja, *parar, necessariamente, para repensar*. Veja-se a descrição da ferramenta na **Estrutura do Ambiente**:

⁶ Tal proximidade parece ser característica da espécie humana, como observam Oeiras e Rocha (2000): "*com o desenvolvimento das redes, o computador passou a ser utilizado também como meio de comunicação. Esse uso se justifica pelo fato que é da natureza humana a necessidade que pessoas têm em se comunicar e que elas são altamente motivadas a interagir qualquer que seja o meio disponível*" Oeiras e Rocha, 2000, p. 1).

Contém materiais que visam desencadear reflexões e discussões entre os participantes ao longo do curso.

Mesmo de posse dessa informação, o formador atribuiu à ferramenta outra funcionalidade que, nos parece, próxima ao *stricto sensu* sugerido pelo seu nome: ele informava os momentos em que haveria parada no sistema por falta de energia, interrupção da comunicação por problemas da rede, etc. E em sua dinâmica deixava claro, dado o momento atual, que os alunos deveriam estar atentos a essas informações caso tivéssemos cortes de energia programados.

O dado mostra a tendência que a nossa língua tem de se relacionar sempre a fatos e acontecimentos provocados pela ação humana [Freire, 1999]. Em outras palavras, o nome da ferramenta leva o formador a relacioná-lo à crise de energia elétrica que vimos atravessando no Brasil. Os cortes de energia anunciados, portanto, estão fortemente ligados a idéia de *parar de fazer alguma coisa obrigatoriamente*, neste caso, as paradas do ambiente TelEduc são noticiadas pelo formador.

Analogamente, convém apresentar outro dado que mostra formas diferentes de se entender e, conseqüentemente, de usar a ferramenta **Diário de bordo**. Projetada, por um lado, para possibilitar ao participante o registro de suas impressões e reflexões a respeito do seu processo de aprendizagem e, por outro, para dar condições ao formador de exercer um acompanhamento mais efetivo e individualizado do processo de cada aluno, nem sempre este propósito é compreendido pelos que dele fazem uso.

Ao que parece, novamente, as (re)significações se assentam, pelo menos em parte, em relações sugeridas pelo próprio nome da ferramenta. A expressão Diário de Bordo pode ser entendida como um conjunto de relatos diários a respeito do que acontece no decorrer de uma jornada qualquer: uma viagem de negócios, uma expedição científica, temporada de férias e, no contexto de EaD, durante um curso. Por outro lado, se apenas a palavra *diário* for considerada, é possível relacioná-la à idéia de se relatar fatos íntimos, confissões, segredos etc.⁷ O nome da ferramenta e sua interface - especialmente o fato de poder ser compartilhado apenas com os formadores - são aspectos que em conjunto contribuem para interpretações discrepantes. Vejamos a primeira delas:

Reivindicações 16/06/2000, 14:29:07

Oi pessoal,

Seria muito bom a troca de experiência entre os grupos de trabalho, e ferramentas disponíveis para compartilharmos é o que não falta (Grupo de Discussão, Diário de Bordo, Correio...). Mas o que venho observando que muitas vezes o grupo cursista não corresponde. As vezes chego a pensar que só tem os formadores no curso. Pois os mesmos quando acionados respondem rapidinho e também no repasse de instruções não deixam a desejar. Já enviei alguns questionamentos a nível de troca de experiência com algumas pessoas do curso, e as vezes digo que estou aguardando o retorno, e nada. Sabemos que o tempo é escasso, mas a troca de experiência é muito válida, podendo ela nortear e enriquecer os nossos próximos planejamentos de atividades com o aluno. Abraços, N.

⁷ Considerando-se sua interface, especialmente o fato de a tela do Diário de Bordo não informar que o que nele estiver registrado será compartilhado apenas com o formador (esta informação encontra-se na Estrutura do Ambiente e no item Sobre... que faz parte de sua tela), não seria surpreendente se algum usuário entendesse a ferramenta como um espaço privado.

A participante reflete sobre a dificuldade de interagir com os colegas e mostra os recursos que utilizou para tentar, sem sucesso, contato. Note-se que ela acha que a ferramenta **Diário de bordo** é compartilhada com todos: sua mensagem é endereçada a todos, alunos e formadores, tanto que inicia o texto com *Oi pessoal*. A participante não se dá conta do funcionamento da ferramenta, possivelmente pela razão de tomá-la como espaço de registro, aos moldes do que dissemos a respeito de um diário de viagem, por exemplo, que não é necessariamente privado no sentido de não compartilhado.

Diferentemente, outros usuários consideram o **Diário de bordo** como *um lugar para dizer o que não seria dito em outro*: um segredo, um cochicho, um assunto particular, semelhante ao que ocorre nas conversas pós-aula entre professor e aluno em tom, quase sempre, reservado, como no exemplo abaixo que trata de uma conversa particular:

Caros Formadores	19/06/2000, 08:11:59
Gostaria da opinião de vocês sobre o seguinte assunto(...).	

Essa forma de interpretar o **Diário de bordo** é confirmada por outro dado, coletado durante o oferecimento de um curso-piloto em que alunos foram transformados em formadores e, portanto, tinham acesso a todas as ferramentas do ambiente, inclusive aos **Diários de bordo** dos colegas. Neste caso, não era mais possível "segredar" com o formador. Foi preciso lançar mão de outra ferramenta que pudesse desempenhar função semelhante. É o que mostra o trecho de uma mensagem colocada no **Correio eletrônico** descrito a seguir:

Dificuldades de trabalho no grupo
(...) Ia colocar este comentário no meu Diário de Bordo, mas não queria que ficasse "aberto" a todos os colegas. (...)

Esses dados sugerem que a atribuição de nomes às ferramentas de EaD, como em todo design de sistemas interativos, merecem atenção especial, levando-se em conta que os nomes constituem pontos de referência iniciais para o usuário dar um sentido e, portanto, uma funcionalidade às ferramentas do ambiente. Mas por outro lado, mesmo com nomes não familiares, os usuários utilizam as ferramentas e as tornam consistentes com as suas intenções de uso. Portanto, diferentemente de outros ambientes computacionais, a escolha "inadequada" de um nome para refletir uma funcionalidade pretendida não conduz a um erro por parte do usuário. E a análise desses usos inusitados contribuem significativamente para um redesign que inclua novas funcionalidades.

4. Considerações Finais

A análise dos dados mostra, dentre outras, a importância de se integrar ao ambiente TelEduc, semelhante ao que já foi feito com a inclusão da ferramenta **InterMap**, novas ferramentas que possibilitem visualizar a *co-presença*. Ferramentas que permitam "dar uma olhada em quem está na sala de aula", como fazemos ao passearmos pelos corredores da universidade, por exemplo. Algumas das estratégias de comunicação desenvolvidas pelos usuários, como a combinação de **Acessos** e **Correio eletrônico** que relatamos é uma solução pouco eficiente, decorrente, possivelmente, de estratégias de navegação que o usuário desenvolve em contato com o ambiente.

Além disso, espaços de conversas síncronas privadas como os "cochichos" dos diários de bordo, e espaços de conversas coordenadas, como as propostas assembléias no bate-papo são essenciais para que se estabeleça formas mais eficazes de formar vínculos

peçoais e com isso fortalecer o processo de formação de uma comunidade de aprendizagem em um curso.

Mas o aspecto mais importante a ser ressaltado é que designers de ambientes para EaD precisam estar atentos a essas inevitáveis (re)significações por parte do usuário e projetar ambientes para EaD implica em termos sistemas onde elas sejam possíveis, pois só assim estaremos focando nosso design na tarefa e no usuário. Não existe a melhor maneira de fazer educação e não se deve ter por objetivo construir ambientes que forcem um determinado caminho ou concepção pedagógica. Portanto, do ponto de vista de IHC, deve-se analisar cuidadosamente essas (re)significações do usuário, não no sentido de que eles nos apontam “falhas” de design que poderiam ser corrigidas via novas terminologias, ou *helps*, ou então treinamento, e sim por elas nos apontarem novas funcionalidades necessárias à tarefa.

Concluindo, esperamos ter mostrado por meio de nossa análise a importância e a especificidade de se projetar ambientes em EaD que suportem constantes (re)significações por parte de usuários diferentes e que, reciprocamente, incorporem - caso pareçam relevantes para a concepção de EaD como um todo - essas (re)significações ao modelo de design. Certamente, isto só é possível em ambientes do tipo aberto, que tratam o usuário como alguém que influencia o ambiente, diferentemente de um *consumidor* que contenta-se em seguir, estritamente, as opções que lhe são apresentadas.

Referências

- Dionísio, A. P. (2001) Análise da Conversação. In: MUSSALIM F. e BENTES, A. C. (Org.) *Introdução à Lingüística: Domínios e Fronteiras*. São Paulo, SP: Cortez. v.2, cap. 3, p. 69-100.
- Freire, F.M.P., Prado, M.E.B.B. (1996) Professores Construcionistas: A Formação em Serviço. *Anais do 3º Congreso Iberoamericano de Informatica Educativa* - Barranquilla, Colombia.
- Freire, F.M.P. (1999) *ENUNCIACÃO E DISCURSO: a linguagem de programação Logo no discurso do afásico*. Campinas: Instituto de Estudos da Linguagem da UNICAMP, 224 p. (Dissertação, Mestrado em Lingüística).
- Goldberg, M. W. (1996) Using a Web-Based Course Authoring Tool to Develop Sophisticated Web-Based Course. Em rede: <http://www.webct.com/service/ViewContent?contentID=11747> [Última consulta: 08/06/2001].
- Harasim, L. et al. (1996) *Learning Networks*. Cambridge, MA: MIT Press, 329 p.
- Haythornthwaite, C. et al. (2000) Community Development Among Distance Learners: Temporal and Technological Dimensions. *Journal of Computer-Mediated Communication*, em rede: <http://www.ascusc.org/jcmc/vol6/issue1/haythornthwaite.html>, v. 6, n. 1.
- Lotus (2001) LearningSpace: A New Vision of e-learning. Em rede: <http://www.lotus.com/home.nsf/tabs/learnspace> [Última consulta: 08/06/2001].
- Maingueneau, D. (1998) *Termos-chave da Análise do Discurso*. Belo Horizonte, MG: Editora da UFMG, 155 p.

- Oeiras, J. Y. Y. e Rocha, H. V. da. (2000) Uma modalidade de comunicação mediada por computador e suas várias interFACES. In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 3, 2000, Gramado. *Anais...* Porto Alegre: Instituto de Informática da UFRGS, p. 151-160.
- Preece, J. (2000) *Online Communities - Designing Usability, supporting sociability*. Chichester: John Wiley & Sons, 439 p.
- Rocha, H.V. e Baranauskas, M.C.C. (2000) Design e Avaliação de Interfaces, Escola de Computação 2000, SBC, IME-USP, SP
- Romani, L. A. S. (2000) *Intermap: Ferramenta para visualização da Interação em Ambientes de Educação a Distância na Web*. Campinas: Instituto de Computação da UNICAMP. 120 p. (Dissertação, Mestrado em Ciência da Computação).
- Romani, L. A. S.; Rocha, H. V. da; Silva, C. G. da (2000) Ambientes para educação a distância baseados na Web: Onde estão as pessoas? In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 3. *Proceedings*. Gramado, Brasil, outubro.
- Valente, J. A. (1998) Por que o computador na educação. In: VALENTE, J. A. (Org.) *Computadores e Conhecimento: Repensando a Educação*. 2ª edição. Campinas, SP: UNICAMP/NIED, cap. 2, p. 29-53.
- Valente, J. A. (1999) Formação de Professores: Diferentes Abordagens Pedagógicas. In: VALENTE, J. A. (Org.) *O Computador na Sociedade do Conhecimento*. Campinas, SP: UNICAMP/NIED, cap. 6, p. 131-156.

Design Ergonômico de Software de Correio e Agenda Corporativos na Web

Antônio Marcos Endler¹, Lia Buarque Macedo de Guimarães² e Flávio Sanson Fogliatto²

¹ Cia. de Processamento de Dados do Estado do Rio Grande do Sul (PROCERGS)
Praça. Açorianos s/nº - CEP: 90010-340 - Porto Alegre – RS

²Programa de Pós-graduação em Engenharia de Produção (PPGEP) – Universidade Federal do Rio Grande do Sul (UFRGS) - Porto Alegre – RS

antonio-endler@procergs.rs.gov.br, (liabmg, ffogliatto)@ppgep.ufrgs.br

Abstract: *This article presents a software graphic interface designing case. This interface concept has been based on ergonomics guidelines and others methods like Macroergonomics Design approach. This research has allowed the identification of the users demands and the measurement of their satisfaction about this. The results have showed that ergonomics guidelines besides others interface concept methods, increment the user's satisfaction rates.*

Resumo: *Este artigo aborda a utilização de diretrizes ergonômicas em conjunto com o método do Design Macroergonômico (DM), proposto por Fogliatto & Guimarães (1999), no desenvolvimento da interface gráfica de um sistema de correio e agenda corporativos. No estudo de caso abordado neste trabalho, o DM permitiu que fossem (i) identificadas demandas dos usuários com relação ao produto e (ii) mensurada a satisfação de usuários e especialistas quanto à qualidade ergonômica do produto desenvolvido. Os resultados obtidos demonstraram que o uso de diretrizes ergonômicas na concepção das telas do produto, usadas de forma integrada com outros métodos, impactaram positivamente nos índices de satisfação dos usuários.*

1. Introdução

Há poucos anos atrás, a interação homem-computador não existia como campo de interesse científico, pois poucas pessoas tinham acesso aos computadores, os quais eram utilizados predominantemente por especialistas. Nos anos 70 e 80, os sistemas eram inoperáveis por pessoas sem competência específica. “A única solução possível envolvia a profunda adaptação do homem ao trabalho com o sistema computacional” (Cybis, 1998, p.50). O surgimento do computador pessoal nos anos 80 e mais recentemente da *Web*, nos anos 90, fez com que o número de usuários crescesse rapidamente. Desta forma, a interação homem-computador (IHC), enfocada na ergonomia de software, torna-se objeto de interesse acadêmico e profissional. Por exemplo, um dos problemas da não ergonomia, apontados por Cybis (1998) são longos tempos de aprendizado, altas taxas de erro, aborrecimentos e frustrações. A ergonomia busca sistemas com o máximo de conforto, segurança e eficiência com o uso de conceitos da psicologia cognitiva, da fisiologia e da análise ergonômica do trabalho.

A PROCERGS (Companhia de Processamento de Dados do Estado do Rio Grande do Sul) vem investindo em pesquisas aplicadas em IHC visando melhorar a usabilidade dos sistemas desenvolvidos. Desde 1996 promove convênios de cooperação técnico-científica com diversas Universidades. O trabalho aqui apresentado é decorrência dos esforços dispensados pela área de pesquisa tecnológica da empresa com o objetivo de criar uma cultura sólida em IHC no processo de desenvolvimento de software.

O objeto de estudo deste trabalho é o *Direto*, um sistema de correio e agenda corporativos, baseado em protocolos abertos, totalmente direcionado ao ambiente *Web*. Este sistema deve ser utilizado por diversas instituições do Governo do Rio Grande do Sul e, portanto, tem importância estratégica para a empresa e para a pesquisa em IHC por abranger usuários com diferentes perfis.

Este artigo aborda o desenvolvimento do projeto *Direto* que utilizou diretrizes ergonômicas e de design gráfico como base para a concepção da interface gráfica do produto. As diretrizes foram aplicadas e posteriormente avaliadas junto aos especialistas no desenvolvimento e aos usuários do produto, através de testes feitos em versões intermediárias denominadas Alfa e Beta. Nestes testes foram empregadas técnicas tradicionais de avaliação de usabilidade do *software* como a Avaliação Heurística e as Listas de Verificação, com posterior aferição dos índices de satisfação do produto através da utilização de formulários especialmente concebidos para o ambiente da *Web*. Os itens de melhoria do produto foram identificados e priorizados com o auxílio do método de Design Macroergonômico (DM; Fogliatto & Guimarães, 1999) que leva em consideração a opinião de usuários e especialistas no processo de desenvolvimento de produtos.

Os procedimentos detalhados de priorização resultantes da aplicação do DM não serão apresentados neste trabalho, pois já foram abordados em outra publicação (Endler, Guimarães e Fogliatto, 2001).

As diretrizes ergonômicas e as diretrizes para o design gráfico da interface foram utilizadas na concepção das telas do produto com a finalidade de evitar os erros mais comuns e produzir uma interface com qualidade funcional, estética e de uso. Essas diretrizes foram utilizadas sempre em conjunto com métodos e técnicas ergonômicas de inspeção (como a avaliação heurística) e técnicas de inquirição (como a prática de entrevistas) nas diversas etapas do desenvolvimento, até ser concluída a primeira versão de testes (versão Alfa) do produto *Direto*. Neste artigo será dado destaque aos procedimentos de mensuração dos índices de satisfação, realizados durante os testes da versão Beta que contou com a participação de usuários da PROCERGS e da METROPLAN (Fundação de Planejamento Metropolitano e Regional).

2. Uso de Diretrizes no Projeto *Direto*

A ausência de aplicação de métodos de concepção de interfaces gráficas que contemplem a participação dos usuários constitui-se um dos principais problemas encontrados no desenvolvimento de software. É importante avaliar se o uso de diretrizes, de métodos de avaliação de usabilidade e de testes com usuários, contribuem para um bom resultado final do projeto da interface, de acordo com a percepção dos usuários. Este artigo pretende mostrar alguns resultados práticos obtidos com o uso de diretrizes que puderam ser mensuradas com o apoio do DM.

Pode-se utilizar diretrizes ergonômicas e de design durante todas as fases do projeto de sistemas. O conhecimento prévio das diretrizes possibilita que os projetistas construam

interfaces com maior qualidade, pois à medida que se projeta conhecendo os critérios de validação e as diretrizes, pode-se evitar os problemas mais comuns de uso. Isso pode reduzir os possíveis retrabalhos nas etapas finais do processo de desenvolvimento, o que acarretaria em atrasos nos cronogramas e aumento dos custos do projeto

Entre as diretrizes ergonômicas mais consistentes pesquisadas estão as diretrizes apresentadas pelo Labiútil/UFSC (1999). Elas estão descritas no site de endereço: <http://www.labiutil.inf.ufsc.br> e podem atuar como apoio ao desenvolvimento e na definição de listas de verificação adotadas em inspeções ergonômicas.

As listas de verificação atuam como instrumentos de avaliação de usabilidade que é um nome genérico para um grupo de métodos baseados na avaliação e inspeção ou no exame relacionado com aspectos de usabilidade da interface com o usuário (Mack & Nielsen, 1992). Os diferentes métodos de inspeção e projeto têm metas diferentes, mas todos eles são uma forma de encontrar problemas no projeto da interface e problemas de usabilidade, enquanto o uso de diretrizes auxilia na concepção das telas, reduzindo o número de problemas encontrados nas avaliações. A avaliação de usabilidade é baseada no julgamento do avaliador. É um processo dinâmico que ocorre durante o ciclo de vida de um produto ou sistema. Quanto mais cedo os testes forem realizados, maiores são as chances de detectar metas pouco claras ou irracionais de usabilidade (Conyer, 1995).

Marcus (1997) destacou que os principais Guias de Interface com Usuários existentes, como os guias dos sistemas operacionais baseados em janelas (o Machintosh GUI - Guide User Interface, o neXTStep GUI e o Microsoft Windows), apresentam recomendações para o projeto de sistemas e para o correto uso dos elementos de interação disponíveis nestes sistemas operacionais. O correto uso destes elementos de interação deve garantir um mínimo de compatibilidade com os diferentes *softwares* existentes no mercado. É importante destacar que muitos dos comportamentos recomendados para os elementos de interação podem funcionar em sistemas na Internet, com o uso de linguagens como o javascript. Contudo, alguns destes elementos de interação são de difícil implementação na Internet no atual estágio da tecnologia. Os desenvolvedores dos *softwares* feitos para funcionar na Internet ainda apresentam dificuldades para implementar o conceito de janela Modal, por isso, se forem abertos diálogos em novas janelas, como ocorre no Windows, o sistema não tem como obrigar o usuário a fechar a janela para prosseguir operando. Os sistemas na Internet também não conseguem dispor da possibilidade do uso de teclas de atalho no teclado para as atividades mais freqüentes, exigindo, muitas vezes, o uso do *mouse*. São limitações da tecnologia empregada que podem afetar os níveis de satisfação dos usuários, relativamente a aspectos ergonômicos.

Diretrizes sobre o projeto gráfico das telas (técnicas de composição e diagramação, do uso de cores, de fontes, de textos, da construção dos ícones, etc) também devem ser consideradas durante a concepção e o desenvolvimento de um sistema, pois impactam diretamente sobre sua usabilidade e função estética. Existem publicações (Rigui, 1993; Parisotto, 1997; Labiútil, 1999) que tratam das questões relacionadas ao uso de diretrizes ergonômicas e de design para a concepção de interfaces gráficas.

Moraes (1998) menciona alguns parâmetros que norteiam uma boa interação homem-computador relacionada ao design gráfico das telas. Deve-se procurar minimizar a quantidade de itens de cada tela, agrupar itens similares, realçar as informações prioritárias, padronizar as telas, dar atenção para as margens, o entrelinhamento e a diagramação dos textos e para o uso das cores.

Com o crescimento da *Web* e do comércio eletrônico, o número de transações realizadas *on-line* multiplica-se e, assim, multiplicam-se os problemas com a usabilidade dos sistemas. A Internet é vista como uma poderosa mídia, capaz de agregar os mais variados serviços. Devido à dimensão econômica que possui, diversos autores têm se dedicado à análise ergonômica das páginas na Internet. Muitas das diretrizes existentes para os sistemas em geral podem ser facilmente adaptadas ao mundo da Internet. Atualmente existem diretrizes específicas para este meio. Entre os autores que elaboraram as mais significativas diretrizes pode-se destacar Nielsen (1996,1999), que desde o surgimento da Internet comercial centraliza sua análise na usabilidade dos produtos e *sites* existentes. Quanto ao posicionamento de Nielsen, é importante destacar que ele é enfático ao afirmar que os usuários preferem que os *sites* funcionem da mesma maneira que outros *sites* já conhecidos e que deve-se simplificar ao máximo os projetos com cuidado especial para a análise das tarefas, para o projeto de conteúdo e para a arquitetura da informação (Nielsen, 2000).

As implicações legais futuras das normas internacionais devem promover uma tendência de que os produtos apresentem rótulos que assegurem a qualidade ergonômica, através da conformidade com o padrão ISO 9241 (Çakir & Dzida, 1997). O uso das normas ISO poderá crescer já que, no Brasil, encontra-se em fase de elaboração desde junho de 1999, os “Requisitos ergonômicos para o trabalho em escritórios com computadores”.

As diretrizes propostas pelo Labiútil (1999) foram utilizadas na concepção da interface do produto Direto, sendo que dois fatores contribuíram para a escolha dessas diretrizes como orientadoras para o projeto. O primeiro fator é que já havia a experiência prévia em validação ergonômica de outro produto da empresa com resultados positivos. E o segundo fator é que as diretrizes adotadas pelo Labiútil são bastante abrangentes e cobrem diversos aspectos sobre usabilidade e utilidade de um *software*. Além disso, a linguagem adotada pelo Labiútil para as diretrizes pode ser considerada de fácil entendimento para não especialistas em ergonomia, facilitando a introdução dos conceitos ergonômicos dentro da empresa. Estas diretrizes serviram de base para a elaboração dos questionários de satisfação e importância aplicados junto aos usuários. É importante destacar que os dados coletados a partir da realização de entrevistas abertas com usuários também foram utilizados como base para a elaboração destes questionários.

3. Design Macroergonômico

Fogliatto & Guimarães (1999) definem demanda ergonômica como manifestações do usuário quanto às questões ergonômicas relativas aos produtos que manuseia e aos postos de trabalho onde executa suas tarefas. Os autores desenvolveram um método que incorpora ferramentas estatísticas de análise de dados e tomada de decisão a uma abordagem macroergonômica e participativa para desenvolvimento de produtos.

O conceito de ergonomia participativa prega a participação dos usuários em todas as fases da intervenção ergonômica. No caso do desenvolvimento de sistemas, deve contemplar, também, a participação dos desenvolvedores de todas as áreas e funções, como o *marketing*, o suporte, os analistas, os programadores e os *designers*. Ao invés de empregar um processo *top down*, unidirecional, onde o ergonômista estuda o problema e recomenda soluções, a ergonomia participativa busca envolver o usuário, garantindo a sua cumplicidade na implantação das soluções. Com isso, a ergonomia participativa transforma os trabalhadores em agentes de melhoria do produto, ao qualificá-lo a identificar e resolver

problemas relacionados ao seu trabalho, muitas vezes dispensando a presença de especialistas (Costella & Guimarães, 1998).

A participação dos usuários na implementação de melhorias em *software*, em especial em *software* de uso geral, faz-se necessária, não apenas como uma maneira de reduzir a resistência natural a mudanças, mas para incorporar os seus conhecimentos ao projeto.

O DM permite discriminar as diferentes demandas dos usuários em função das suas atividades, a partir de estratégias de coleta de informação que privilegiam a sua opinião. A aplicação de ferramentas estatísticas possibilita a priorização das demandas e o estabelecimento de relações entre estas e suas possíveis soluções. Até o momento, o DM foi utilizado com sucesso em projetos de postos de trabalho (Guimarães *et al.*, 1998; Fogliatto & Guimarães, 1999; Krug, 1999) e ambientes de trabalho (Van der Linden, 1999; Ramirez, 2000).

A metodologia de DM é operacionalizada através do uso de ferramentas para seleção de amostras e coleta de dados, como questionários e entrevistas estruturadas, assim como estratégias para organização das informações obtidas. As opiniões e desejos manifestados pelos usuários são processados com base em um conjunto de técnicas estatísticas e de tomada de decisão, gerando dados confiáveis para elaboração de parâmetros ergonômicos de projeto. Esses dados são consolidados como características ou itens desejados pelo usuário diante das necessidades de sua tarefa ou de uso do produto em estudo. No DM, essas características são denominadas Itens de Demanda Ergonômica (IDEs).

A implementação do DM segue um processo estruturado em sete etapas; as seis primeiras etapas destinam-se a gerar parâmetros de projeto, enquanto que a última etapa corresponde à interação com a atividade projetual propriamente dita. As sete etapas são:

- 1) Identificação do usuário e coleta organizada de informações.
- 2) Priorização dos Itens de Demanda Ergonômica (IDEs) identificados pelo usuário.
- 3) Incorporação da opinião de especialistas.
- 4) Listagem dos Itens de *Design*.
- 5) Determinação da força de relação entre Itens de Demanda Ergonômica e Itens de *Design*.
- 6) Tratamento ergonômico dos Itens de *Design*.
- 7) Implementação do novo *design* e acompanhamento.

4. Construção dos questionários eletrônicos

As etapas 1 e 2 do DM definiram os IDEs, onde observou-se a sua direta relação com as diretrizes publicadas pelo Labiútil (1999). As diretrizes, juntamente com os dados coletados a partir da realização de entrevistas abertas, foram a base para a construção dos questionários eletrônicos que avaliaram os graus de satisfação e importância dos usuários. Por este motivo, serão detalhadas apenas as etapas 1 e 2 do DM. Na etapa 1 (Identificação do usuário e coleta organizada de informações) foi realizada uma entrevista aberta para a definição dos itens a serem incluídos no questionário.

A entrevista procurou não induzir os usuários, conforme proposto no DM; foram aplicadas três perguntas abertas e abrangentes. As entrevistas foram conduzidas com 9 usuários, tendo sido gravadas com um gravador portátil. As entrevistas abrangeram usuários que, na sua maioria, já possuíam contato com algum *software* de correio eletrônico e agenda corporativos. A entrevista foi constituída por três questões:

- 1) O que você espera de um *software*?
- 2) O que você espera de um *software* de Correio e Agenda?
- 3) O que você não gosta no *software* de correio e agenda que você utiliza?

Os 9 usuários selecionados para as entrevistas possuíam diferentes funções organizacionais e níveis de experiência com este tipo de ferramenta. Entre os entrevistados estavam secretárias, gerentes, técnicos em computação, técnicos administrativos, *designers*, atendentes de *helpdesk* e clientes da empresa.

Após a conclusão das entrevistas, foi feita uma compilação das respostas identificando a sua ordem de menção, o que foi utilizado como um fator de ponderação na análise dos resultados.

Na etapa 2 (Priorização dos Itens de Demanda Ergonômica (IDEs) identificados pelo usuário) procurou-se valorizar os primeiros itens mencionados nas entrevistas. Assim, o primeiro item recebeu peso $1/1=1$, o segundo $1/2=0,5$, o terceiro, $1/3=0,33$. A partir do quarto item citado, todos receberam a pontuação de $1/4=0,25$. Itens mencionados por mais de um entrevistado tiveram seus índices somados. Os itens ergonômicos presentes em listas de verificação conhecidas na literatura, como as listas propostas pelo Labiútil (Laboratório de Utilizabilidade da Universidade Federal de Santa Catarina; 1999), foram comparados com os itens demandados pelos usuários. Foi constatado que a maioria dos itens demandados pelos usuários eram apresentados nas listas de verificação. Os especialistas mantiveram os principais itens ergonômicos presentes nas listas de verificação, agregando itens adicionais elencados e não constantes nas listagens.

A partir das entrevistas foram elaborados dois questionários: um sobre o grau de satisfação (foco principal de análise deste artigo) acerca dos IDEs e outro sobre o grau de importância atribuído aos IDEs. Os dois questionários continham 24 questões e mais questões complementares para a estratificação da amostra, como: nome do usuário, sexo, idade, grau de instrução, empresa que trabalha, função exercida, tempo de empresa, tempo de uso da Internet, e questões referentes ao uso de *software* de correio e agenda. Os procedimentos adotados para o desenvolvimento dos dois questionários (importância e satisfação) foram os mesmos: programação em *html* e armazenamento dos dados no banco *Postgress*. Os dois questionários foram aplicados durante os testes da versão Beta do produto, que contou com 70 usuários selecionados na PROCERGS e na METROPLAN. Os participantes dos testes responderam o questionário sobre o grau de importância no primeiro dia de uso do programa e responderam o questionário sobre o grau de satisfação após duas semanas de uso do programa.

Os participantes dos testes receberam uma mensagem eletrônica com a indicação dos prazos-limite para responder o questionário. Esta mesma mensagem apresentou um *link* (elemento básico de interação da *Web* que leva o usuário para algum destino específico) denominado "avaliação do grau de satisfação". Com um clique do usuário no *link*, era aberta uma nova janela com o questionário. No final do formulário foram incluídos os botões limpar (que limpa todas as respostas) e enviar (que envia automaticamente o

5. Resultados

O enunciado resumido das questões e os índices de satisfação resultantes dos testes da versão Beta realizados junto aos usuários podem ser observados na Tabela 01.

Questões	Índice de Satisfação (%)
q12 - Entendimento dos ícones	86,59
q36 - Adequação da carga de informações com a capacidade de memorização do usuário	86,50
q11 - Aspecto visual das telas do produto	86,46
q16 - Facilidade de leitura das informações	85,88
q14 - Uniformidade visual de uma tela para outra em contextos idênticos	85,73
q33 - Aproveitamento dos conhecimentos adquiridos com produtos similares e outros softwares	85,61
q13 - Organização visual dos itens de informação	84,75
q24 - Possibilidade de acesso ao produto a partir de outros microcomputadores	84,38
q414 - Objetividade dos textos empregados	83,50
q15 - Destaque visual para os itens e funções mais relevantes	81,83
q34 - Tempo de leitura das telas	78,75
q31 - Facilidade de uso do produto	78,50
q21 - Número de funções existentes no produto	77,63
q38 - Facilidade de acesso ao ajuda do sistema	77,24
q32 - Informações recebidas sobre o uso do produto	76,95
q22 - Adequação das funções existentes às necessidades	75,50
q411 - Número de passos necessários para a realização das atividades	73,88
q37 - Facilidade de acesso às opções de personalização/configuração	73,38
q39 - Qualidade das informações fornecidas no Ajuda	72,88
q410 - Tempo de resposta para a exibição de novas telas	70,24
q413 - Possibilidade de realizar uma atividade de diferentes maneiras	69,76
q412 - Controle sobre a atividade	68,25
q23 - Opções de personalização/configuração disponíveis	67,88
q35 - Confiança de que o software executa é aquilo que é esperado	59,88

Tabela 01 – Índices de Satisfação dos usuários do Direto em ordem decrescente

Do total de 70 usuários selecionados e cadastrados nos testes, 53 responderam os questionários. Destes, 35 responderam os questionários de importância e satisfação. Esperava-se um número maior de usuários da Metroplan, mas a instituição designou um número reduzido de usuários (12) e, destes, 7 responderam pelo menos um dos questionários. Na PROCERGS foram selecionados 58 usuários e destes, 46 responderam pelo menos um dos questionários, indicando uma participação efetiva por parte dos usuários nos testes. A maioria dos usuários enviou um número significativo de mensagens eletrônicas para os administradores (número difícil de ser ponderado, pois elas foram enviadas também para vários desenvolvedores) contendo descrição de problemas, dificuldades e sugestões de melhoria que também contribuíram na elaboração da lista de melhorias.

Dos que responderam o questionário sobre o grau de importância, 14 eram mulheres e 30 eram homens. A idade média dos usuários que responderam este questionário ficou em 34,9 anos. O tempo médio de empresa foi de 8,7 anos e o tempo médio de uso de Internet e correio eletrônico foi, respectivamente, de 4,4 e 6,2 anos. Analisando esses índices médios, pode-se constatar que o perfil dos usuários da amostra é, na maioria de, usuários experientes com ferramentas de correio eletrônico. A idade média dos usuários que responderam o questionário de satisfação variou pouco em relação ao de grau de importância, ficando em 35,4 anos; destes, 13 eram mulheres e 28 eram homens. O tempo médio de empresa foi de 7,8 anos e o tempo médio de uso de Internet e correio eletrônico foi, respectivamente, de 4,3 e 5,9 anos. Dentro destas empresas, foi difícil identificar e selecionar participantes para os testes com pouco tempo de uso em alguma ferramenta de

correio eletrônico, principalmente na PROCERGS que apresenta um nível de informatização dos processos bastante elevado.

Muitos dos usuários que não responderam os questionários encontravam-se de férias ou viajando no período dos testes. Outros alegaram falta de tempo. Contudo, a participação dos usuários pode ser considerada positiva, considerando que os testes foram realizados no ambiente de trabalho concorrendo com as atividades normais de cada um, além de exigir algumas horas de dedicação no período de 2 semanas.

Foram priorizadas as diretrizes que, na avaliação dos desenvolvedores, eram viáveis dentro do contexto tecnológico. As diretrizes ergonômicas propostas pelo Labiútil (1999) apoiaram diretamente a concepção dos ícones (Q12), a distribuição dos itens de informação (Q13) e a busca pela consistência das telas (Q14). Os índices de satisfação dos usuários nesses itens foram, respectivamente, 86,59%, 84,75% e 85,73%. O uso de diretrizes de design (Parisotto, 1997; Righi, 1993) auxiliou na definição do design gráfico das telas, que na questão Q11 - "aspecto visual das telas", apresentou o terceiro índice de satisfação mais elevado, 86,46%, em relação a todos os IDEs.

As questões avaliaram muitas das diretrizes ergonômicas propostas pelo Labiútil (1999) que, após as entrevistas com os usuários, mostraram-se abrangentes e de acordo com o que os usuários esperavam de um sistema. Somente quatro novas questões foram incluídas nos questionários aplicados na versão Beta.

A questão Q33, relacionada com a diretriz ergonômica de compatibilidade proposta pelo Labiútil (1999) apresentou um dos índices mais elevados de satisfação (85,61%).

A maioria dos *bugs* poderia ter sido corrigida antes dos testes da versão Beta. Isso ficou evidenciado pelo baixo índice de satisfação dos usuários na questão Q35, relacionada à confiabilidade do *software*, que apresentou um índice de 59,88%. Muitas funcionalidades apresentaram problemas que podem ter influenciado na percepção dos usuários em relação a outros IDEs. Por isso, a correção e a identificação dos *bugs* foi priorizada pela equipe de desenvolvimento a partir desses resultados.

6. Conclusão

O uso de diretrizes (Labiútil, 1999; Marcus, 1997; Nielsen, 1996, 1999, 2000; Parisotto, 1997; Righi, 1993) apoiou a concepção da usabilidade e do *design* gráfico do sistema, contribuindo para o resultado final do projeto, impactando nos índices de satisfação dos usuários e especialistas. Contudo, somente a intenção de seguir diretrizes não garante resultados práticos do ponto de vista do usuário fazendo-se necessários procedimentos de validação e testes dos sistemas para detectar implementações incorretas, possibilitando uma melhora na qualidade percebida pelo usuário.

As diretrizes utilizadas pelo Labiútil (1999) mostraram-se abrangentes em relação às demandas dos usuários, aferidas a partir da entrevista aberta. A maioria dessas diretrizes pôde ser utilizada tanto na lista de verificação ergonômica como nos questionários sobre o grau de importância e o grau de satisfação, sendo necessárias apenas algumas adaptações de linguagem aos usuários e aos especialistas, já que estavam de acordo com as demandas dos usuários.

Os baixos índices de satisfação encontrados em alguns IDEs (Q410 - tempo de resposta para exibição de novas telas e Q412 - controle sobre a atividade) podem ter relação com as limitações da tecnologia da Internet, já que o produto apresentou desempenho similar às páginas mais rápidas da rede, com tempo médio de 6 segundos para o carregamento das telas. Porém, os dados coletados não são suficientes para comprovar

essa relação. Mesmo assim, existem duas soluções viáveis para melhorar os índices: desenvolver um sistema em ambiente Windows para ser instalado na máquina do usuário, ou melhorar a infra-estrutura da rede, através do aumento de velocidade e da capacidade de tráfego de informações, dependendo das condições de operação da Rede Estado RS (rede de alta velocidade para transmissão dos dados do RS).

O baixo índice mensurado na pesquisa com os usuários em relação à confiabilidade do software, representado na questão Q35, refere-se a um problema relacionado diretamente com a área de informática mas que influencia diretamente na percepção dos usuários quanto à usabilidade. Isso demonstra que os métodos de concepção e avaliação ergonômica devem estar integrados aos métodos existentes de desenvolvimento de sistemas. Essa necessidade de integração ficou evidente neste trabalho.

A implementação de determinados elementos de interação (Marcus, 1997), como as janelas modais e o uso de teclas de atalho, não foi possível, devido às restrições impostas pelo atual estágio de desenvolvimento das tecnologias empregadas. Para usuários que utilizam o produto intensivamente, como as secretárias de diretoria, as teclas de atalho são muito importantes; por isso, a equipe de desenvolvimento deve continuar pesquisando uma forma de incorporar esta característica ao produto.

A equipe de desenvolvimento mostrou-se comprometida com a aplicação dos métodos ergonômicos no projeto. Sem o comprometimento das equipes de desenvolvimento, pode-se perder muito da eficiência dos métodos, pois de nada adianta a realização de análises e a realização de testes com usuários se a equipe desenvolvedora do produto não está comprometida com os resultados. O DM foi bem recebido pelos desenvolvedores, pois desde o início “escutou” a voz dos usuários e dos desenvolvedores, criando um comprometimento com as soluções de melhoria.

O método de Design Macroergonômico mostrou-se bastante flexível no desenvolvimento do produto *Direto* e permitiu a priorização dos itens de melhoria levando em consideração a opinião dos usuários, sem deixar de considerar a opinião dos especialistas. O DM atuou como uma importante ferramenta para a administração do produto, possibilitando que a equipe de desenvolvimento tomasse decisões de projeto baseadas em critérios ponderáveis e tangíveis.

O uso de versões intermediárias de teste do produto permitiu a sua avaliação em diferentes circunstâncias, garantindo um bom padrão de qualidade ergonômica. A expectativa dos desenvolvedores é que esses procedimentos tragam, como consequência, a aceitação e a satisfação em relação ao produto por parte dos usuários. Contudo, a aplicação dos testes da versão Beta do produto demonstraram a insatisfação dos usuários a respeito da confiabilidade do sistema. Isto está associado ao grande número de problemas de mal funcionamento do produto, que poderiam ter sido corrigidos antes da aplicação destes testes. Para que isso não comprometesse o processo de implantação, decidiu-se alocar recursos humanos específicos no processo de verificação de *bugs*, onde cada correção deverá ser acompanhada e testada novamente, até que o produto apresente uma confiabilidade satisfatória.

Os questionários eletrônicos desenvolvidos mostraram-se uma boa alternativa para a aplicação de pesquisas em instituições com bom grau de informatização. Não foi relatado, por parte dos usuários e dos desenvolvedores, nenhum comentário negativo sobre

os questionários. O envio de mensagens constantes pelo correio eletrônico solicitando a participação nos testes, colaborou para que o número de respondentes fosse maior.

Por fim, o trabalho aqui apresentado possibilitou uma adequação dos conhecimentos acadêmicos e teóricos à prática projetual, aplicada no desenvolvimento de um *software*, consolidando uma cultura dentro da empresa que privilegia as questões de IHC.

7. Referências Bibliográficas

- ÇAKIR, Ahmet, e DZIDA, Wolfgang. International Ergonomic HCI standards – Handbook of Human Computer Interaction In HELANDER, M. 2ED Elsevier, 1997.
- CYBIS, Walter A. Abordagem Ergonômica para o Desenvolvimento de Sistemas Interativos. In: P&D DESIGN, 1998, Rio de Janeiro. Anais. Rio de Janeiro: Estudos em Design/AEND-BR. pag0049-0057.
- CONYER, Merle. User and Usability testing – how it should be undertaken? Australian Journal of Educational Technology, 1995, 11(2), 38-51.
- COSTELLA, Marcelo.; GUIMARÃES, Lia B. M.; CREMONINI, Ruy A. Análise dos acidentes do trabalho ocorridos na atividade de construção civil no Rio Grande do Sul em 1996 e 1997. In: XVIII Encontro Nacional De Engenharia De Produção, 1998, Niteroi, RJ. Anais do Encontro Nacional de Engenharia de Produção. 1998.
- FOGLIATTO, Flávio.; GUIMARÃES, Lia B.M. Design Macroergonômico: uma proposta metodológica para projeto de produto. Produto & Produção Porto Alegre, v.3, n.3, 1999.
- GUIMARÃES, Lia B.M.; LEAL, Andreia F. N.; FISCHER, Daniela et al. Recomendações ergonômicas para posto de pedágio: Relatório de pesquisa. Porto Alegre: PGP/UFGRS, 1998.
- KRUG, Sérgio. Aplicação do método de Design Macroergonômico no projeto de postos de trabalho: estudo de caso de posto de calibração. Porto Alegre, 1999. Dissertação de mestrado em Engenharia de Produção – UFRGS.
- LABIUTIL/UFSC - Laboratório de Utilizabilidade/Universidade Federal de Santa Catarina. Web ErgoList. <http://www.labiutil.inf.ufsc.br/ergolist>, (documento recuperado em 29/04/1999).
- MACK, Robert L, e NIELSEN, Jakob. Usability inspection methods: rapport on a workshop held at chi'92. Monterey, CA, maio,1992. SIGCHI Bulletin, pag. 28-33.
- MARCUS, Aeron. Graphical User interfaces. In HELANDER, M. Hand book of Human Computer-Interaction. 2ed. Elsevier, 1997. pag 423-440.
- MITCHELL, M. & JOLLEY, J., 1996. Research Design Explained. HBJ College, New York.
- MORAES, Anamaria Ergonomia e Interação Homem-Computador, usabilidade de interfaces: linha de pesquisa. In: P&D DESIGN, 1998, Rio de Janeiro. Anais. Rio de Janeiro: Estudos em Design/AEND-BR, pag0038-0042.

- NIELSEN, Jakob. The Top Ten Mistakes in Web design. <http://www.useit.com/alertbox/9605.html>, maio,1996 (documento recuperado em 17/04/2000).
- NIELSEN, Jakob. The Top Ten New Mistakes of Web design. <http://www.useit.com/alertbox/990530.html>, maio,1999 (documento recuperado em 28/08/2000).
- NIELSEN, Jakob. Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis, Indiana, USA, 2000.
- PARIZOTTO, Rosamelia. Elaboração de um Guia de Estilos para Serviços de Informação em Ciência e Tecnologia via Web. Dissertação de Mestrado, PPGE/UFSC - Programa de Pós-graduação em engenharia de produção/Universidade Federal de Santa Catarina, Florianópolis, 1997.
- RAMIRES, A. M. Análise de acidentes radiológicos no contexto organizacional das empresas de radiologia industrial. Dissertação de Mestrado, PPGE/UFGRS - Programa de Pós-graduação em engenharia de produção/Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- RIGHI, Carlos A. R. Aplicação de Recomendações Ergonômicas ao Componente de Apresentação da Interface de Softwares Interativos. Dissertação de Mestrado, PPGE/UFSC - Programa de Pós-graduação em engenharia de produção/Universidade Federal de Santa Catarina, Florianópolis, 1993.
- VAN DER LINDEN, Júlio C. S. Identificação dos Itens de Demanda Ergonômica em Escritório Informatizado. Dissertação de Mestrado, PPGE/UFGRS - Programa de Pós-graduação em engenharia de produção/Universidade Federal do Rio Grande do Sul, Porto Alegre, 1999.

Estudo e Avaliação da Usabilidade de Agentes Improvisacionais de Interface

Márcia Cristina Moraes¹, Ana Carolina Bertoletti², Antônio C. da Rocha Costa³

¹PPGC – UFRGS e FACIN – PUCRS
Av. Bento Gonçalves, 9500 – Porto Alegre – RS – Brazil

²Museu de Ciências e Tecnologia – PUCRS
Av. Ipiranga 6681 – Prédio 40 – Porto Alegre – RS – Brazil

³Escola de Informática – UCPel
Félix da Cunha, 412 – Pelotas – RS – Brazil

mmoraes@inf.ufrgs.br, carolina@upf.tche.br, rocha@atlas.ucpel.tche.br

***Abstract.** In this paper we are going to present a special kind of software agents, that work on systems' interfaces, namely, the improvisational agents. This kind of agent has attracted attention of researchers in Artificial Intelligence and Human-Computer Interaction because they can be designed in order to be capable to motivate and stimulate users in their use of systems. This motivation occurs because such agents can vary their actions and behaviors in an improvised, but coherent, way. We are going to present a theoretical revision of improvisational agents and the results of an usability evaluation process that we made to the improvisational agents developed for the SAGRES Virtual Museum.*

***Resumo.** Neste artigo apresentaremos um tipo de agente voltado para interfaces, o agente improvisacional. Este agente tem atraído a atenção de pesquisadores nas áreas de Inteligência Artificial e Interfaces Homem-Computador pois possibilita o projeto de agentes capazes de motivar e estimular usuários ao uso de sistemas. Esta motivação e estímulo ocorrem na medida em que o agente pode improvisar ações e comportamentos de forma coerente. Mostraremos uma revisão teórica dos conceitos envolvidos com agentes improvisacionais de interface e a avaliação da usabilidade dos agentes improvisacionais desenvolvidos no Museu Virtual SAGRES.*

1. Introdução

Segundo Hayes-Roth (1995) e Chacra (1983), a improvisação está presente em diversas atividades humanas como teatro, música e dança. Até mesmo as atividades consideradas mais formais como lecionar, apresentar trabalhos e defender propostas e teses possuem algum grau de improvisação. Nunca sabemos ao certo quais surpresas e questionamentos teremos que enfrentar ao realizar uma destas atividades, e a nossa eficiência será medida em relação a nossa capacidade de saber lidar com estas situações. De fato, a vida é repleta de situações inesperadas e constantemente temos que ajustar nossos planos para seguir em frente em busca do nosso objetivo. Certamente, se os seres humanos, os agentes mais

inteligentes do mundo, utilizam esta capacidade de adaptação e improvisação, é porque ela é interessante e até mesmo fundamental para o desenvolvimento da vida em sociedade.

Se a improvisação é algo importante para os agentes humanos, por que não utilizá-la em agentes que habitam ambientes computacionais? Os ambientes computacionais, assim como o mundo real, podem ser bastante dinâmicos e complexos, pois simulam situações reais e permitem a interação com agentes humanos. A Inteligência Artificial tem proposto o desenvolvimento de agentes de software para habitar ambientes com estas características. Estes agentes normalmente consideram aspectos tradicionalmente tratados na Inteligência Artificial como planejamento, raciocínio e aprendizagem. Contudo, se os ambientes são dinâmicos, eles podem apresentar surpresas aos agentes, e para tratar destes imprevistos os agentes devem possuir capacidades de improvisação.

A necessidade da capacidade de improvisação é mais evidente quando os agentes são representados através de personagens animados ou figuras e faces humanas que interagem com o usuário. Algumas pesquisas mostram que os usuários aplicam regras sociais aos computadores, mesmo que suas interfaces não sejam explicitamente antropomórficas (Ball 1997) (Koda 1996a) (Koda 1996b). Desta maneira, nada mais interessante do que personificar a interface incorporando aos agentes uma representação humana ou de personagens. Assim como os agentes humanos, os agentes de software representados graficamente devem apresentar comportamentos coerentes, interessantes e ao mesmo tempo credíveis. Para agir de forma credível, o agente de software deve simular comportamentos dos agentes humanos. Sendo assim, o agente de software deve possuir amplas capacidades e uma destas capacidades é a improvisação.

Deste modo, este artigo apresenta a utilização da improvisação na Inteligência Artificial mais especificamente em agentes de software voltados para a interface com o usuário, os chamados agentes improvisacionais. O artigo inicialmente apresenta uma revisão literária que mostra a importância da personificação nos agentes de interface e o papel que a improvisação de comportamentos tem neste processo. Logo após são apresentados alguns conceitos relacionados aos agentes improvisacionais como, definição, requisitos e critérios de avaliação.

Vários pesquisadores têm realizado avaliações de agentes de interface personificados. Neste artigo iremos apresentar uma avaliação para agentes improvisacionais de interface, iremos validar os agentes improvisacionais desenvolvidos para o Museu Virtual SAGRES, do Museu de Ciências e Tecnologia da Pontifícia Universidade Católica do Rio Grande do Sul (MCT/PUCRS). O Museu Virtual SAGRES tem como objetivo buscar uma cooperação entre museus e escolas a fim de criar um novo ambiente educacional que possibilite a educação continuada e disponibilize acesso as informações do museu para a comunidade em geral. Neste sentido, a interface do SAGRES deve ser interessante e de fácil entendimento para diferentes tipos de usuários. Os agentes foram aplicados ao sistema para atender estes requisitos e proporcionar ajuda aos usuários. Através da avaliação dos agentes improvisacionais do SAGRES pretendemos verificar se os agentes cumprem o seu papel e se a improvisação de comportamentos os torna ao mesmo tempo atraentes e úteis para os usuários.

2. Representação Gráfica e Antropomorfismo em Agentes de Interfaces

Segundo Elliott (1998), as pessoas são animais sociais e como tal possuem habilidades sociais altamente desenvolvidas como fala e conversação. Contudo, estas habilidades não estão sendo utilizadas para gerenciar o uso e obter informações das aplicações de computadores. Atualmente os usuários têm que se adaptar aos protocolos de comunicação artificiais estabelecidos entre eles e as aplicações. Elliott (1998) salienta que o progresso das interfaces está em reduzir as necessidades de adaptação dos usuários e aumentar a habilidade do programa se comunicar dentro de um contexto social humano. Esta é a área onde os programas baseados em agentes, em especial os agentes de interface que aparecem para o usuário como personagens dramáticos animados e exibem qualidades sociais é particularmente atrativa.

Os agentes de interface são atrativos pois tem como objetivo suportar e fornecer assistência para um usuário aprender uma aplicação em particular. O agente observa e monitora as ações tomadas pelo usuário, aprendendo e sugerindo maneiras mais eficientes de realizar uma tarefa. Desta maneira, o agente age como um assistente pessoal autônomo, que coopera com o usuário para realizar alguma tarefa na aplicação. Para interagir com o usuário são necessárias algumas habilidades sociais como: manter histórico do seu relacionamento com o usuário, ter conhecimento das emoções humanas, entender a fala humana e falar com o usuário.

Complementando as idéias de Elliott, Singh e Huhns (1997) afirmam que a animação de componentes é um exemplo de uma tendência mais geral na engenharia de software que é construir software que imitam objetos do mundo real. Segundo os autores, se esta imitação é bem feita, o software irá parecer familiar e conseqüentemente fácil de usar. Seguindo esta linha de raciocínio, algumas pesquisas em agentes de interface têm considerado importante incluir uma representação gráfica aos agentes, seja esta na forma de personagens dramáticos animados (Elliot 1998) (Hayes-Roth 1998) ou na forma humana (Koda 1996a). O consenso é que a "personificação/antropomorfismo" das interfaces, incluindo comportamentos e expressões faciais semelhantes aos dos seres humanos, auxiliam o usuário a se engajar na tarefa, criando um relacionamento baseado na familiaridade, afeto e confiança (Hayes-Roth 1998) (Koda 1996a).

Laurel (1997) salienta que a familiaridade que as pessoas possuem em interagir com personagens na vida real (amigos, família, animais de estimação) e a capacidade de inferir ações futuras dos personagens baseando-se nas suas características internas e externas é que possibilita a personificação das interfaces com o usuário. Desta maneira, ao projetar agentes de interface, as características externas, como dicção e aparência devem ser modeladas para sugerir as características internas, como valores e heurísticas, que determinam como o personagem vai fazer escolhas e executar ações. A integração das características internas e externas é responsável pela criação de um personagem coerente, que possui a capacidade de selecionar cuidadosamente os comportamentos que devem ser executados a cada momento, criando a ilusão de vida no personagem ou forma humana.

Várias nomenclaturas têm sido aplicadas para agentes que possuem comportamento coerente e criam uma ilusão de vida ao personagem ou forma humana. Estes agentes têm sido chamados de personagens sintéticos, agentes sintéticos, agentes credíveis, agentes pedagógicos, etc. Acreditamos que todos estes agentes apresentam uma característica comum, que é explícita ou implicitamente representada, a capacidade de improvisação.

Desta maneira, iremos chamar os agentes que criam a ilusão de vida ao interagir com os humanos de agentes improvisacionais de interface.

3. Agentes Improvisacionais de Interface

A idéia que geralmente se faz a respeito da improvisação é de algo informal, espontâneo, imprevisível, sem preparo prévio, inventado de repente, aleatório, enfim trata-se de um produto inspirado na própria ocasião e feito sem preparo. Em oposição tem-se a idéia do “ser em forma” como algo preparado, organizado, elaborado, deliberado, portanto um produto formalizado. No primeiro caso costuma-se inserir a improvisação e no segundo a representação (Chacra 1983). Contudo, estas definições não implicam que improvisação e representação sejam duas coisas distintas, elas são, na realidade, dois pólos de uma mesma matéria. A distância entre estes dois pólos, o improvisado e o representado, é que determina as diferenças entre si através de graus, onde uma apresentação se torna mais ou menos formalizada ou improvisada (Moraes 2001).

Esta noção de improvisação foi aplicada na Inteligência Artificial para a criação dos agentes improvisacionais. Agentes improvisacionais são um tipo específico de agentes de software voltados para a interface que utilizam o paradigma da improvisação dirigida (Hayes-Roth 1997) para interagir com o usuário.

O paradigma da improvisação dirigida pressupõe a existência de um mundo virtual habitado por atores animados. Estes atores animados são na realidade agentes cuja função é interagir com o usuário, a fim de que juntos realizem alguma tarefa. Hayes-Roth e Doyle (1998) definem atores animados como criaturas de computador com mentes de software, que se assemelham aos atores reais e fictícios numa série de dimensões qualitativas e funcionais. Por exemplo, eles possuem funções cognitivas básicas para percepção e ação dentro do seu contexto da aplicação, expressam seus comportamentos através de diferentes maneiras (como movimentos, gestos e conversas) e diferentes mídias (como gráficos, animações, voz e texto), podem possuir qualidades psicológicas como personalidade, emoção, motivação e relacionamentos sociais, podem possuir diferentes tipos de conhecimento, e utilizá-los de diferentes maneiras dentro da aplicação, e podem encenar diferentes papéis em diferentes tipos de aplicações.

Os atores envolvidos na improvisação dirigida são definidos por três construtores dramáticos: enredo, papel e personagem (Hayes-Roth 1997). Enredo é uma seqüência de ações temporariamente restringida envolvendo um conjunto de indivíduos. Papel é a classe de indivíduos cujos comportamentos, relações e interações são conhecidas tanto dos atores quanto da audiência. O personagem representa uma personalidade definida como uma configuração coerente de traços psicológicos. São os personagens, com suas personalidades distintas, que provocam reações emocionais nas pessoas e que as fazem amá-los ou odiá-los. O enredo, o papel e o personagem são substancialmente independentes e podem ser dirigidos ou improvisados separadamente. Isto significa que os atores podem ser dirigidos em qualquer um destes conjuntos e deixados livres para improvisação em outro. Apesar dos construtores serem substancialmente independentes consideramos que, um ator (agente) pode interpretar um ou mais personagens e cada personagem pode estar relacionado a um ou mais papéis que serão assumidos no momento da execução. Desta maneira, os atores estão indiretamente relacionados aos papéis que os seus personagens executam. A figura 1 mostra um exemplo de ator animado e seus construtores.

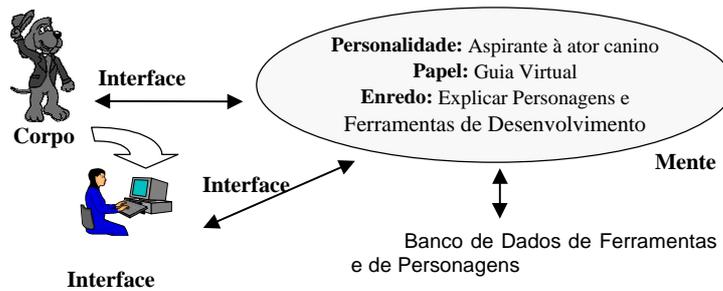


Figura 1: Exemplo de Ator Animado com seus Construtores

Na improvisação dirigida, os atores trabalham dentro de restrições de direções que variam o grau com o qual elas restringem o comportamento dos atores. As direções podem ser abstratas e deixar o ator quase que completamente livre ou podem ser bastante específicas, descrevendo os menores detalhes de comportamento. Elas podem restringir qualquer aspecto do comportamento incluindo o papel, caracterização e ações do ator. As direções podem ser informadas na forma de cenários completos (enredo, papel e personagem) ou podem ser informadas interativamente, durante a execução. Estas direções podem vir de diversas fontes, incluindo pessoas, outros componentes de computador ou os próprios atores. Desta maneira, os atores seguem instruções recebidas e ao mesmo tempo surpreendem com comportamento habilmente improvisado.

3.1. Requisitos de um Agente Improvisacional

Para que um ator possa ser considerado um agente improvisacional, é necessário que ele atenda aos seguintes requisitos Hayes-Roth (1995) e Hayes-Roth e Doyle(1998):

- possuir uma personalidade: os agentes improvisacionais não se caracterizam somente pela sua função específica, mas pelo que eles incorporam à execução de sua função, como por exemplo identidade visual e qualidades distintas como personalidade e psicologia;
- possuir empatia: eles devem responder e perceber adequadamente os sentimentos dos usuários e devem mostrar seus sentimentos de maneira coerente. O que se espera é que os agentes se comportem como se percebessem, sentissem e manifestassem emoções;
- possuir relações sociais: segundo Hayes-Roth e Doyle (1998), Reeves e Nass observaram que as interações dos indivíduos com computadores são fundamentalmente social e natural. Desta maneira, os agentes devem ser projetados para evocar respostas sociais desejáveis. Estas respostas estão relacionadas a comportamentos sociais que incluem aparência, qualidade da voz, estilo de conversação, linguagem corporal, etiqueta e dinâmica emocional;
- ter capacidades funcionais de percepção, raciocínio e ação em ambientes em tempo-real complexos e incertos.;
- exibir inteligência cotidiana, ou seja, possuir um senso comum sobre o mundo físico, um modelo psicológico simples e convenções sociais;

- exibir comportamento vivo, ou seja, poder variar a interpretação de uma determinada instrução dependendo da ocasião, poder executar comportamentos específicos em diferentes ocasiões, poder ter emoções associadas com seu comportamento e interações;
- colaborar com outros personagens e compartilhar controle;
- exibir perícia improvisacional: a improvisação dirigida é explicitamente orientada a processos, ou seja, a instrução de comportamento fornecida ao ator/agente, por meio do seu enredo, papel e personalidade, é o seu objetivo. Não se pode dizer que a sua execução está “correta” ou “incorreta”, pois ele pode executar de diferentes maneiras as restrições das instruções dos usuários, sendo que todas levam a uma execução “correta”.

3.2. Critérios de Avaliações dos Agentes Improvisacionais

Assim como os requisitos para um agente ser considerado improvisacional são diferentes dos requisitos que caracterizam os agentes tradicionais da Inteligência Artificial, os critérios de avaliação aplicados aos agentes improvisacionais devem ser diferentes dos critérios de avaliação dos agentes tradicionais.

Hayes-Roth e Doyle (1998) salientam que os trabalhos em agentes inteligentes herdam os critérios de avaliação da Inteligência Artificial e de outros campos da Ciência da Computação envolvidos no projeto de agentes, como Interfaces Homem-Computador. Estes critérios definem as qualidades desejáveis dos sistemas. Contudo, se espera que os agentes improvisacionais se comportem de maneira semelhante aos humanos e como consequência os critérios tradicionais de avaliação devem ser adaptados ou modificados. Deste modo, Hayes-Roth e Doyle (1998) apresentam uma adaptação de alguns critérios considerados importantes para avaliação de agentes representados através de personagens animados. Os critérios são:

- confiabilidade se torna variabilidade: o comportamento dos agentes tradicionais, deve ser confiável, ou seja, eles devem fazer a mesma coisa todas as vezes que operam sob um dado conjunto de circunstâncias. Contudo, para ser semelhante aos humanos e credíveis os agentes improvisacionais devem possuir variabilidade na escolha a maneira de executar determinados comportamentos;
- previsibilidade se torna idiosincrasia: o comportamento dos agentes tradicionais, deve ser previsível, deve-se poder saber como ele irá executar em circunstâncias críticas. Contudo, como as pessoas e os personagens fictícios, os agentes improvisacionais devem trocar padrões previsíveis de comportamento por surpresas interessantes;
- o correto se torna apropriado: os agentes tradicionais devem executar corretamente. As pessoas são criadas para executar tarefas importantes e os erros não são aceitáveis, mas errar é humano. Os personagens devem agir apropriadamente, dados seus papéis, suas circunstâncias e natureza imperfeita semelhante a dos humanos;
- o completo se torna efetivo: com poucas exceções, a maioria das pessoas encontra-se sempre num estado de aperfeiçoar suas habilidades, possuindo um conhecimento incompleto que precisa ser melhorado. Da mesma maneira, os agentes

improvisacionais devem funcionar efetivamente, algumas vezes com conhecimento incompleto;

- eficiência se torna interesse: os agentes tradicionais devem executar de maneira eficiente. Contudo, os agentes improvisacionais devem abrir mão da eficiência em favor de caminhos que intrinsecamente são mais interessantes;
- ótimo se torna distinção individual: o mais alto grau de sucesso para os agentes tradicionais é definido como a execução ótima, onde o agente atinge o sucesso máximo em cada um dos critérios discutidos anteriormente. Para as pessoas e os personagens fictícios, a execução ótima é indefinida e irrelevante. O que se procura nas pessoas e nos personagens é a distinção individual cujos comportamentos surpreendem as pessoas e são marca registradas de cada indivíduo.

A fim de avaliar a efetividade e impacto dos agentes improvisacionais sobre usuários reais, analisamos os agentes improvisacionais presentes no Museu Virtual SAGRES do MCT/PUCRS. Estes agentes foram desenvolvidos utilizando-se a arquitetura e estrutura de agentes improvisacionais propostas em Moraes (Moraes 1999a) e possuem todos os requisitos necessários para sua classificação como agentes improvisacionais.

4. Museu Virtual SAGRES

O SAGRES é um ambiente educacional construído na Web que facilita a organização de visitas a museus, apresentando as bases de informações dos museus de forma adaptada as características dos usuários (capacidades e preferências). O sistema determina o grupo de links apropriados para o(s) usuário(s), mostrando-os em uma página HTML resultante (Bertoletti e Costa 1999).

O sistema permite três tipos de usuários: visitante, professor e aprendiz. O visitante é responsável por construir e gerenciar sua visita ao museu. Ele pode escolher quais assuntos irá estudar e quais atividades irá realizar. O professor é responsável por construir uma visita sobre algum assunto e a tornar disponível para um grupo de estudantes. O aprendiz é capaz de realizar uma visita previamente planejada por um professor.

O SAGRES possui um módulo de agentes responsável por gerar e gerenciar guias virtuais. Estes guias são agentes improvisacionais representados através de personagens animados que tem como função auxiliar os usuários durante a navegação e operação do sistema além de analisar e monitorar as ações dos usuários. Como mencionado anteriormente os agentes do SAGRES foram desenvolvidos utilizando-se a metodologia apresentada em Moraes (Moraes 1999) e possuem todos os requisitos para serem classificados como agentes improvisacionais. Sendo assim, os agentes podem decidir o que fazer e como se comportar durante a sua interação com o usuário. Para fazer isto os agentes possuem *scripts* de ações que contém as instruções que eles devem executar e os possíveis comportamentos relacionados a elas. Os comportamentos possíveis são selecionados dentro de um conjunto de comportamentos considerando o contexto da ação de interação com o usuário. Desta maneira, uma instrução pode ser executada por diferentes comportamentos e a escolha de qual executar é feita em tempo de execução de maneira improvisada (aleatória). Estes comportamentos incluem comportamentos físicos (movimentos, gestos e animações) e verbais (falas) similares aos dos seres humanos, como acenar, vibrar e saudar. Com esta representação nós acreditamos estar oferecendo uma interface mais amigável para os usuários. Para verificar a validade de nossa afirmação na seção 5 apresentamos a

avaliação realizada sobre os agentes do SAGRES. As figuras 2, 3 e 4 mostram exemplos dos agentes no SAGRES.

Para suportar a independência entre o modo dos agentes e o SAGRES nós adotamos o modelo de desenvolvimento em três camadas: apresentação, negócios e dados. As camadas de dados e negócios estão localizadas no servidor. A camada de dados armazena os bancos de dados do SAGRES e informações relativas aos agentes. A camada de negócio implementa as funcionalidades dos agentes. A camada de apresentação está localizada no lado cliente e contém as páginas HTML do SAGRES e as páginas dos agentes. O modelo em camadas pode ser visualizado na figura 5.



Figura 2: Entrada no Sistema



Figura 3: Página de Escolha de Tipo de Usuário



Figura 4: Opções de Consulta

5. Avaliação dos Agentes Improvisacionais do Museu Virtual SAGRES

Para que um sistema tenha sucesso e seja efetivamente utilizado, é necessário que ele seja útil para um determinado público alvo, seja fácil de aprender e manipular e que a sua interface seja atrativa, chamando a atenção do usuário e o estimulando. Neste sentido, a avaliação da usabilidade de interfaces é uma etapa importante dentro do ciclo de vida de um sistema e seu objetivo é identificar problemas que possam comprometer a interação do usuário com a interface.

Existem atualmente vários métodos para avaliação da usabilidade. Estes métodos são classificados como métodos de inspeção da usabilidade e testes empíricos com os usuários. Alguns autores salientam que para se realizar uma análise mais abrangente do software podem ser aplicadas mais de uma técnica a uma mesma interface (Nielsen e Mack 1994)(Winckler et al. 2000).

Neste artigo utilizamos testes com os usuários realizados através de um questionário para avaliar a usabilidade dos agentes improvisacionais do SAGRES, desta maneira as próximas seções irão apresentar os aspectos relacionados a esta avaliação. Informações relacionadas as avaliações com questionário da usabilidade geral do sistema e teste de inspeção usando o ErgoList, *checklist* específico para inspeção de conformidades ergonômicas, podem ser obtidas em (Moraes et al 2001).

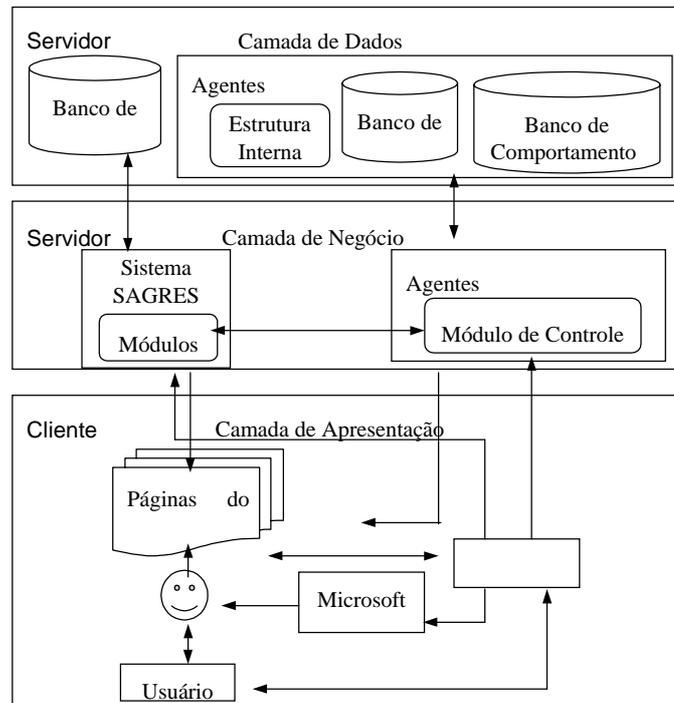


Figura 5: Modelo de Três Camadas do SAGRES

5.1. Teste com Usuários

Medeiros e Cybis (2000) realizaram uma análise de diferentes trabalhos para avaliação da usabilidade e constataram que uma das maneiras de se medir a usabilidade é através da identificação do grau de satisfação dos usuários, no que se refere aos princípios de diálogo previstos na parte 10 da norma ISO 9241.

A fim de verificar o grau de satisfação e conseqüentemente a usabilidade do sistema SAGRES, foi desenvolvido um questionário para avaliar a usabilidade dos agentes improvisacionais. O questionário possui perguntas organizadas conforme as categorias dos princípios de diálogo da ISO 9241-10: Adequação à Tarefa, Auto-descrição, Controlabilidade, Conformidade com as Expectativas do Usuário, Tolerância a Erros, Suporte a Individualização, Adequação ao Aprendizado (Blanchard 1997).

Segundo as características das sete categorias apresentadas em Medeiros e Cybis (2000) e considerando as mudanças nos critérios de avaliação para agentes de interface propostos por Hayes-Roth e Doyle (1998), adaptamos os princípios de diálogo da ISO 9241-10 para que a mesma suportasse as modificações propostas. Desta maneira, visualizamos as sete categorias como: adequação à tarefa (avalia se o agente fornece ao usuário o suporte necessário para que as tarefas sejam realizadas com sucesso e se

apresenta comportamentos apropriados a execução das tarefas); auto-descrição (avalia se o agente fornece assistência ao usuário a fim de tornar o programa mais compreensível); controlabilidade (avalia se o usuário pode iniciar e controlar as ações realizadas pelo agente); conformidade com as expectativas dos usuários (avalia se o agente supre as expectativas do usuário em relação a auxílio na realização de tarefas e busca por informação sem causar distração); tolerância a erros (avalia se o agente atinge os resultados apesar da ocorrência de erros nas entradas fornecidas pelos usuários); suporte a individualização (avalia se o agente trata o usuário de maneira personalizada e se apresenta variabilidade de comportamentos para uma mesma tarefa); adequação ao aprendizado (avalia se o agente auxilia o usuário durante a navegação e uso do sistema, fornecendo informações relevantes para o aprendizado do mesmo. Para realizar esta função o agente deve assumir o papel de guia do usuário, acompanhamento o usuário durante toda a sua interação com o sistema).

O critério de tolerância a erros não foi considerado nesta avaliação, pois o agente não recebe nenhuma entrada do usuário. As informações que o agente possui em relação ao usuário são transmitidas diretamente através dos módulos do SAGRES.

Cada questão do questionário tem quatro alternativas de respostas: ruim, regular, bom e muito bom. Cada um destes conceitos foi associado a um peso de um à quatro, como ilustrado na tabela 1. A única exceção é a última questão do questionário dos agentes, onde o usuário deve escolher entre as opções com agente ou sem agente. Algumas das perguntas presentes no questionário dos agentes são: *O guia apresentou diferentes comportamentos durante a execução de tarefas específicas?* e *O guia foi útil para auxiliá-lo na operação do sistema?*.

Questão	Ruim	Regular	Bom	Muito Bom
O guia apresentou diferentes comportamentos durante a execução de tarefas específicas?	Peso 1	Peso 2	Peso 3	Peso 4

Tabela 1: Formato das Questões

5.2. Identificação da Amostra de Usuários

Para realização dos testes com usuário foram selecionados alunos de primeiro, segundo e terceiro grau que fazem parte do Clube do Computador do Museu de Ciências e Tecnologia da PUCRS. Esta seleção foi realizada de forma aleatória, sendo que os alunos envolvidos são de ambos os sexos, com diferentes níveis de habilidade e conhecimento, sendo todos usuários assíduos do sistema.

5.3. Análise dos Resultados

Com base nos dados obtidos através dos testes com usuários, considerando a adaptação dos princípios de diálogo para agentes improvisacionais, verificamos que a média do questionário foi de 3,68 e que a maioria dos critérios ultrapassou este valor. Os critérios obtiveram os seguintes percentuais: auto-descrição (3,81), suporte a individualização (3,75), adequação a tarefa (3,71), controlabilidade (3,71), adequação ao aprendizado (3,7) e conformidade com as expectativas do usuário (3,37).

Constatamos que os usuários se mostraram favoráveis a utilização do guia (agente improvisacional), pois o mesmo os auxiliou durante a operação do sistema, tratando-os de maneira personalizada e incentivando-os na realização de tarefas, o que pode ser observado

pelos critérios de auto-descrição e suporte a individualização. O guia atingiu, de maneira bastante favorável, os critérios propostos por Hayes-Roth e Doyle, mostrando ter comportamento distinto, variado e apropriado em diferentes situações e ser atrativo e interessante, segundo o critério de adequação a tarefa.

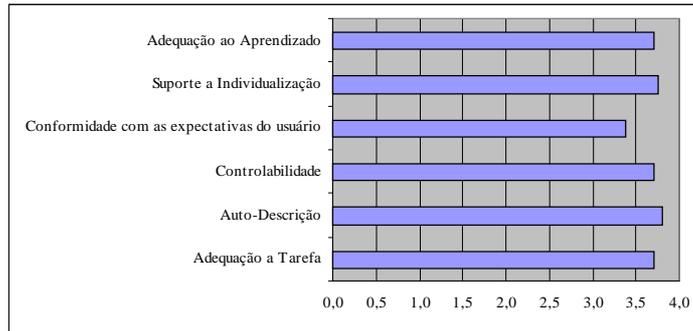


Figura 6: Média dos dados obtidos no questionário

O critério de conformidade com as expectativas do usuário foi o de menor percentual (3,37). Acreditamos que isto tenha ocorrido em virtude do guia ser um personagem animado que pode realizar movimentos (animações) mesmo quando se encontra no estado ocioso, ou seja, quando não é requisitado pelo usuário. Para termos certeza de que este é um motivo da baixa conformidade com as expectativas do usuário iremos realizar uma nova avaliação da usabilidade.

Concluimos que os usuários se mostraram satisfeitos com o uso do guia virtual. Esta satisfação fica explícita quando os usuários foram questionados quanto a sua preferência de uso do sistema com ou sem guia, sendo que todos optaram por realizar suas próximas visitas com o guia virtual. Desta maneira pode-se constatar que o guia virtual foi importante para auxiliar os aprendizes durante a operação do sistema, cumprindo sua principal funcionalidade e serviu de fonte incentivadora para futuras utilizações do sistema.

6. Considerações Finais

Este artigo teve como propósito dar uma visão ampla da área de agentes improvisacionais de interface mostrando sua parte teórica e prática através da avaliação dos agentes improvisacionais presentes no sistema SAGRES.

Vários autores têm realizado estudos para verificar a utilidade da personificação dos agentes de interface. Ball et al. (1997) constaram em seus estudos que a metáfora social incluída com a presença do personagem na interface reduz a ansiedade associada ao uso do computador. Koda (1996a) (1996b) concluíram em seu experimento que a representação do agente traz um maior engajamento do usuário à tarefa, fazendo com que o usuário preste mais atenção ao interagir com o sistema. Na avaliação apresentada também constatamos que os usuários se sentem mais engajados na realização de tarefas, pois os guias os incentivam e os auxiliam fornecendo ajuda personalizada.

Com o questionário podemos verificar que os usuários preferem operar o sistema usando o guia virtual. Desta maneira, concluímos que os agentes improvisacionais oferecem uma abordagem interessante para a implementação de interfaces mais atrativas e personalizadas, aumentando o grau de satisfação do usuário e conseqüentemente a usabilidade de sistemas.

7. Referências

- Ball, G.; et al. (1997) Lifelike Computer Characters: The Persona Project at Microsoft. In: *Software Agents*. Menlo Park, California: AAAI Press.
- Bertoletti, A.C.; Costa, A.C.R. (1999) SAGRES – A Virtual Museum. In: *Museums and The Web 1999 Conference*, D. Bearman and J. Trant (eds.). New Orleans, Louisiana.
- Blanchard, H.E. (1997) Standards: User Interface Standards in the ISO Ergonomics Technical Committee. In: *SIGCHI Bulletin*, vol.29, n.1, January.
- Chacra, S.(1983) *Natureza e Sentido da Improvisação Teatral*. São Paulo. 118p.
- Elliott, C.; Brzezinski, J.(1998) Autonomous Agents as Synthetic Characters. *AI Magazine*. New York, NY: AAAI Press. Pp. 13-30.
- Hayes-Roth, B.; Brownston, L. Sincoff, E. (1995) *Directed Improvisation by Computer Characters*. Technical Report (KSL-95-04) – Stanford University.
- Hayes-Roth, B.; Van Gent, R. (1997) Story-Making with Improvisational Puppets. In: *AGENTS'97*. Disponível por WWW em: <http://sigart.acm.org/proceedings/agents97>.
- Hayes-Roth, B.; Doyle, P. (1998) Animate Characters. In: *Autonomous Agents and Multi-Agent Systems*, v. 1, pp. 195-230.
- Koda, T.; Maes, P. (1996a) *Agents with Faces: The Effect of Personification*. Disponível por WWW em: <http://tomoko.www.media.mit.edu/groups/agents/>.
- Koda, T. (1996b) *Agents with Faces: A Study on the Effects of Personification of Software Agents*. MIT. Dissertação de Mestrado. 1996.
- Laurel, B. (1997) Interface Agents: Metaphors with Character. In: *Software Agents*. California: AAAI Press.
- Medeiros, M.A.; Cybis, W.A. (2000) Método de Avaliação de usabilidade de *software* a partir da satisfação de usuários e da aplicação de quesitos da norma ISO 9241. In: *IHC'2000 – III Workshop sobre Fatores Humanos em Sistemas de Computação*.
- Moraes, M. C. (1999) *Uma Arquitetura de Agentes Improvisacionais para Apoio a Visitação de Museus Baseada em Computadores*. Porto Alegre: Mestrado em Informática PUCRS. 130f. Dissertação de Mestrado.
- Moraes, M. C. (2001) *Agentes Improvisacionais: Uma Perspectiva para Agentes de Software Voltados a Interface*. Porto Alegre: PPGC/UFRGS.Exame de Qualificação.
- Moraes, M. C.; Bertoletti, A.C.; Costa, A.C.R. (2001) *Evaluating Usability of SAGRES Virtual Museum: Considering Ergonomics Aspects and Virtual Guides*. In: 7th World Conference on Computers in Education.
- Nielsen, J.; Mack, R. (1994) *Usability Inspection Methods*. John Wiley & Sons, Inc.
- Singh, M. P.; Huhns, M. N. (1997) Internet-Based Agents: Applications and Infrastructure. *IEEE Internet Computing*. Vol. 1,n.4, pp. 8-9.
- Winckler, M.; Nemetz, F.; Lima, V. (2000) Interação entre Aprendiz e Computador: Métodos para Desenvolvimento e Avaliação de Interfaces. In: *Tecnologia Digital na Educação*. Editora da UFRGS. pp. 7-33.

Gerando conhecimento sobre os homens, mulheres e crianças que usam computadores: algumas contribuições da psicologia clínica

Ana Maria Nicolaci-da-Costa¹, Carla Faria Leitão¹, Daniela Romão-Dias¹

¹ Departamento de Psicologia – PUC-Rio - Rua Marquês de São Vicente, 225 – Gávea
22543-900 Rio de Janeiro, RJ

anicol@psi.puc-rio.br, carlaleitao@uol.com.br, danirom@gb1.com.br

Abstract: *The development of friendly software and interfaces is demanding close knowledge of characteristics (such as likes and dislikes, preferences, aversions, desires, etc.) of different types of users. Given that these characteristics are difficult to capture by means of quantitative methods, a methodology frequently used in clinical psychology is presented to the HCI public: one in which the data collected in open-question interviews are analysed qualitatively.*

Resumo: *O desenvolvimento de software e interfaces amigáveis está demandando o conhecimento próximo de características (como gostos, preferências, aversões, desejos, etc.) de diferentes tipos de usuários. Dado que essas características são de difícil captação por meio de métodos quantitativos, uma metodologia muito utilizada na área de psicologia clínica é apresentada ao público de IHC: aquela na qual os dados levantados em entrevistas de perguntas abertas são analisados qualitativamente.*

1. Introdução

Muitos imaginam que a psicologia clínica está sempre associada às práticas de consultório. Isso não é fato, no entanto. A psicologia clínica comporta várias áreas de investigação. Uma delas é a dos chamados estudos da subjetividade, que procuram tornar visíveis e compreensíveis os processos de construção e o funcionamento de diferentes “configurações psicológicas” (ou seja, de diferentes formas de ser e estar no mundo).

Tradicionalmente, em praticamente todas as suas vertentes, a psicologia clínica manteve-se distante das áreas tecnológicas. Estudar o ser humano não envolvia estudar o seu uso das tecnologias disponíveis. E isso tinha uma razão de ser. Até o final do século XX, acontecia uma de duas coisas: ou as tecnologias disponíveis eram muito básicas, ou seus impactos sobre os seres humanos – mesmo quando bastante profundos como no caso da invenção da imprensa – eram diluídos no tempo. Ainda era cedo para se falar generalizadamente de uma interação dos seres humanos com as máquinas das quais se utilizavam.

Nas últimas décadas, no entanto, esse quadro foi profundamente alterado. A vertiginosa difusão dos microcomputadores e da Internet fez com que milhões de homens, mulheres e crianças ao redor do mundo passassem a interagir cotidianamente com seus computadores pessoais e a reagir a essa interação de diferentes modos. Os impactos psicológicos desse contato do ser humano leigo com as tecnologias digitais foram tão profundos que intimaram a psicologia clínica a se aproximar da informática e a informática

a se interessar pelo conhecimento do ser humano gerado pela psicologia clínica (a esse respeito, ver Leitão e Nicolaci-da-Costa, no prelo).

Essa não é, no entanto, uma aproximação fácil. Muito pelo contrário, é uma aproximação que demanda grande dose de esforço, tanto por parte dos profissionais da psicologia quanto dos da informática. Há, como todos sabemos, grandes diferenças entre suas formas de pensar e de trabalhar. Para mencionar apenas algumas, são diferentes suas concepções de homem (universal *versus* socialmente construído), é diferente sua atribuição de valor ao que é objetivo ou subjetivo, é diferente a sua forma de encarar o espontâneo (no sentido de não pensado), o irracional e o racional, são diferentes suas metodologias, etc.

Essas diferenças geram desconfianças nos profissionais de ambas as áreas. Para dar somente um exemplo, no caso que conhecemos de perto – o da psicologia – a desconfiança em relação àquilo que vem das áreas tecnológicas é tão grande que gerou uma visão negativa das tecnologias digitais difícil de combater. Ainda hoje, muitos clínicos revelam sofrer de “tecnofobia” (ver, por exemplo, Birman, 1985). Isso porque consideram que a utilização cotidiana das tecnologias digitais vem transformando homens, mulheres e crianças em seres cada vez menos “humanos”.

Essa certamente não é a visão que temos. Partimos da premissa de que as tecnologias digitais, tal como outras tecnologias, não são boas ou más em si mesmas. Sabemos, no entanto, que todas as tecnologias que são realmente revolucionárias geram impactos imprevistos sobre os seres humanos. Por isso mesmo, desde os primórdios da difusão dos microcomputadores e da Internet no Brasil, vimos nos dedicando a investigar:

(a) Como os homens, mulheres e crianças contemporâneos estão reagindo às novas tecnologias (gostos, preferências, aversões, desejos, etc.). Ver, por exemplo, Nicolaci-da-Costa (1998).

(b) Como usam as novas tecnologias (uso esse que pode ser bastante diferente daquele previsto por seus criadores). Ver, por exemplo, Nicolaci-da-Costa (2000) e Zaremba, Abreu e Nicolaci-da-Costa (2000).

(c) Quais os impactos que seu uso das novas tecnologias tem sobre suas vidas (relações pessoais, profissionais, formas de falar, de pensar, de produzir, de se agrupar, de se comunicar, de se informar, de sentir, etc.). Ver, por exemplo, Costa (2001), Romão-Dias (2001) e Zaremba (2001).

Partindo da experiência obtida nesses e em outros estudos, pretendemos, neste artigo, dar uma pequena contribuição para o estabelecimento de um contato proveitoso entre a informática e a psicologia clínica. Para tanto, concentraremos nossa atenção em diferentes formas de gerar conhecimento sobre os grandes articuladores da aproximação dessas duas áreas: os homens, mulheres e crianças que usam os computadores e a Internet.

2. Entrando em contato com novos usuários

2.1. Alfabetizando os usuários leigos

Enquanto o computador era principalmente uma ferramenta de trabalho dos profissionais das áreas tecnológicas ou exatas, seu uso necessitava de conhecimentos específicos amplamente dominados por aqueles que o operavam. Engenheiros, físicos, matemáticos,

etc., além de programadores e analistas de sistemas, tinham a habilidade técnica e o raciocínio lógico necessários para usá-lo com relativo conforto.

Em um dado momento, no entanto, “nasceram” os usuários leigos; usuários que, apesar de todo o desenvolvimento tecnológico ocorrido no século XX, sequer se davam conta de que já tinham contato com diferentes tipos de tecnologia; usuários cuja experiência cotidiana informava que máquinas eram coisas fáceis de operar.

Esses usuários nasceram da curiosidade ou de contingências (geralmente associadas ao trabalho). Homens, mulheres ou crianças, não importa, eles resolveram enfrentar o desafio gerado pelo desconhecimento e se embrenhar no novo mundo dos computadores.

Enquanto esses usuários eram pouco numerosos, não eram problema. Por serem desbravadores, eles eram suficientemente curiosos para suportar um processo de “alfabetização” nas linguagens digitais. Eles conseguiram, a duras penas, aprender a escrever o que, aos seus olhos, eram estranhos códigos. Ao contrário dos usuários originais – os profissionais da informática e áreas afins –, no entanto, esses usuários leigos não conheciam a lógica subjacente à digitação dos comandos operacionais. Sentiam-se, portanto, extremamente desconfortáveis ao executar tarefas cuja razão de ser não compreendiam.

Podemos dizer, resumidamente, que o “nascimento” de uma população pouco numerosa de usuários leigos gerou um tipo de relação homem-computador calcada na adaptação dos usuários ao funcionamento de equipamentos e programas. Os profissionais da informática, ainda inexperientes no contato com esses novos usuários, buscaram transmitir-lhes aquela parte de seus conhecimentos – a linguagem dos comandos – que era imprescindível para operar um computador. Dado que não podiam transmitir a lógica por trás desses comandos para leigos, terminaram levando a cabo o que poderia ser chamado de uma “alfabetização mecanizada do usuário”. Esta rapidamente veio a ser questionada dentro da própria informática.

2.2. Conquistando os usuários leigos

Com o *boom* tecnológico e com a economia e o mercado voltados para as novas tecnologias computacionais, o número de usuários leigos começou a se multiplicar rapidamente. A partir daí, tanto na área da pesquisa acadêmica quanto na da pesquisa comercial, os profissionais da informática passaram a se preocupar em tornar as tecnologias digitais mais acessíveis aos usuários cada vez mais numerosos. Como revela Michael Dertouzos (1997), diretor do Laboratório de Ciência da Computação do MIT (Massachusetts Institute of Technology), o principal desafio da informática desde a sua difusão passou a ser o de “humanizar” a tecnologia, ou seja, o de adequar as tecnologias aos usuários em vez de tentar adaptá-los àquelas.

Os resultados desse processo de humanização foram muito positivos. Fazendo uso de dados de pesquisa, intuição, imaginação, criatividade e muito ensaio-e-erro, como revelam os depoimentos do polêmico Bill Gates (1995), passou a ser desenvolvido software para uso leigo. Com isso foram criados recursos mais confortáveis e simples para aqueles que nada entendiam de programação. Bons exemplos desse novo tipo de software são os programas para *Windows*, cujas interfaces gráficas tornaram os comandos operacionais invisíveis.

A tentativa de adoção da ótica de uso de homens, mulheres e crianças comuns também possibilitou a criação de vários outros recursos, úteis não somente para leigos, dos quais seguem alguns exemplos bastante conhecidos do grande público: os corretores ortográficos e gramaticais dos editores de texto, os vários tipos de correção automática que passaram a ser colocados como *default* e as instruções de uso disponíveis *online* no *help* dos programas.

Esses são poucos exemplos do muito que foi feito. São suficientes, no entanto, para mostrar que o grande esforço dos profissionais da informática foi bem-sucedido. A vida do usuário foi facilitada e um número cada vez maior de pessoas foi convidado a estabelecer uma relação amigável – ou, pelo menos mais amigável do que antes – com suas máquinas inteligentes.

Apesar dessa constatação irrefutável, ainda havia muito trabalho pela frente. Isso ficou claro quando os usuários leigos, já familiarizados com os novos recursos que lhes foram oferecidos, passaram a externar suas insatisfações.

2.3. Muitos esforços, muitas conquistas, muitas insatisfações

Os usuários leigos logo perceberam que, apesar de mais fácil, seu relacionamento com computadores continuava sendo muito difícil. E, para a surpresa de muitos profissionais, algumas de suas insatisfações diziam respeito aos próprios recursos que haviam sido criados para ajudá-los. Vejamos alguns exemplos.

Os já mencionados corretores ortográficos dos editores de texto trouxeram muitas facilidades mas também muitos inconvenientes. O MSWord 2000, por exemplo, tem um corretor que erra a acentuação de algumas palavras. Um desses erros pode ser observado quando o usuário digita a forma contraída do verbo fazer acompanhado de seu objeto direto, como em “fazê-lo”. O corretor detecta o que vê como “erro” e sugere que a ortografia correta é “faze-lo”, sem acento. Mais do que isso, essa sugestão consta no *default* do programa que, caso habilitado, retira automaticamente o acento. Para um usuário familiarizado com o uso de computadores, é simples a operação de desabilitar o *default*. Um iniciante (provavelmente em quem se pensava quando foi feito o *default*), no entanto, se surpreende (e se irrita) com a intrusão e não sabe o que fazer quando o acento é retirado automaticamente.

Poderíamos pensar que, frente a esse tipo de ocorrência, o usuário iniciante recorrerá ao *help* do editor de textos. A utilidade desse mecanismo de auxílio é, no entanto, questionável. Os usuários parecem ter muita dificuldade em usar o *help* (do editor de textos e de qualquer outro aplicativo). Para eles, o *help* talvez seja tão complicado quanto o próprio programa. Isso é o que sugere uma pequena enquete que fizemos no ano de 2000. Dos 23 usuários que responderam às nossas perguntas, apenas três disseram fazer uso muito ocasional do *help*. Seguem-se duas de suas respostas.

- Você sabe que estou apenas "engatinhando" nessa história de computador... Eu uso pouco o recurso de ajuda... acho que nem sei usá-lo direito. (mulher, 44 anos, enfermeira)
- Olha, minha resposta é não, não uso o *help*. Além de ser fraquinho, quando você está realmente precisando de ajuda, ele só enrola a sua vida! :) (mulher, 20 anos, estudante)

No mundo dos programas de *e-mail* e dos *browsers* a confusão não é menor. Apesar de os programadores terem conseguido, em poucos anos, desenvolver software de comunicação e navegação bem mais simples do que aqueles dos primeiros tempos da Internet, os usuários ainda se sentem muito perdidos. As reações destes deixam claras as grandes dificuldades que ainda encontram ao tentar usar equipamentos e programas. O *e-mail* que se segue, enviado por uma usuária leiga, brasileira, residente na França, é um bom exemplo disso:

Recebemos uma resposta sua (...) mas nao conseguimos ler? ver? ouvir? o documento anexo. Quando tentamos abrir aparece um ponto de exclamacao dizendo que o documento nao pode ser aberto pela razao 31, o que quer que isso signifique. (...) Continuando: nao sabemos usar (nem explicar) quase nada do que se nos oferece. Sei dizer que estamos usando o Netscape (suponho que isto seja o navegador). Temos tambem o Internet Explorer (que eu suponho ser o navegador do Bill), mas nao usamos por incompetencia, por preguiça ou ainda por medo.

Fica evidente que, por mais simples e óbvias que as interfaces, configurações e instruções tentem ser, a vida de um iniciante na Internet ainda é bastante difícil. Se acrescentarmos aos percalços no uso dos programas de navegação e de comunicação, todas as dificuldades de uso dos demais programas mencionados acima (e de muitos outros) veremos que usuários e programadores ainda têm muitos obstáculos pela frente.

Dertouzos apontava os problemas existentes no ano de 1997 por meio do seguinte comentário bem-humorado:

Na última década, qualquer pessoa que pronunciou a expressão *user friendly* na minha presença correu o risco de agressão física. Ela foi usada descaradamente para sugerir que um programa é fácil e simples de operar, quando isso raras vezes corresponde à verdade. (p. 327)

Desde então, é óbvio que muito já foi feito para tornar prazerosa – ou, ao menos, não penosa – a experiência de usar um computador. É óbvio, também, que ainda há muito por fazer. Um passo importante foi dado. A estratégia de humanização do computador deu bons resultados iniciais. Não foi, porém, suficiente. Está cada vez mais claro que, para o desenvolvimento de software e interfaces realmente amigáveis, é necessário conhecer gostos, preferências, hábitos, opiniões, sugestões, medos, aversões, desejos, etc. dos seres humanos que os usam.

3. Conhecendo os usuários leigos

Conhecer os seres humanos a partir de diferentes ângulos tem sido a tarefa prioritária das ciências humanas. Por isso mesmo, elas desenvolveram vários métodos para coletar e analisar dados. Os métodos mais freqüentes de coleta de dados são as diversas variações da observação, em ambientes naturais ou em laboratório, e as pesquisas nas quais homens, mulheres e/ou crianças fornecem explicitamente informações sobre si mesmos por meio de respostas às perguntas que lhes são feitas em questionários ou entrevistas. Já os métodos de análise de dados se dividem em basicamente dois: os quantitativos e os qualitativos.

Tendo em vista a complexidade que uma discussão metodológica pode assumir nas ciências humanas, concentraremos nossa discussão em dois tipos de pesquisa. O primeiro,

muito usado na área de psicologia experimental, caracteriza-se pelo uso de questionários fechados para a coleta de dados e de métodos quantitativos para a análise do material coletado. O segundo, muito usado na área de psicologia clínica, faz uso de entrevistas para a coleta de dados e de métodos qualitativos para a análise destes. Para facilitar a referência, chamaremos os primeiros de pesquisas quantitativas e os segundos de pesquisas qualitativas.

3.1 A pesquisa quantitativa e a identificação de perfis de usuários

Durante muito tempo as pesquisas quantitativas dominaram o cenário da psicologia. Seu pressuposto básico é um raciocínio estatístico bastante conhecido: o de que os resultados obtidos a partir de uma amostra estatisticamente representativa de uma população podem ser generalizados para toda essa população¹.

Este raciocínio, que está por trás de várias pesquisas dentro da psicologia, também está por trás de outras pesquisas que são amplamente divulgadas pela mídia. As últimas são obviamente de mais fácil acesso aos leitores deste artigo. Por isso mesmo, foi dentre elas que escolhemos um exemplo: o de uma grande pesquisa sobre perfis de usuários realizada por um renomado instituto de pesquisa, o Ibope (em associação com o Cadê?). Esperamos que esta dê maior visibilidade à nossa argumentação.

Em um trecho do *e-mail* de divulgação das conclusões da quarta versão da pesquisa Cadê/Ibope, realizada no ano de 1999 e divulgada em 2000, podemos observar que os resultados tiveram origem nos dados fornecidos por uma ampla amostra de 25 mil usuários. (São, no entanto, raras as pesquisas quantitativas que exibem uma amostra dessa magnitude.)

A pesquisa Cadê?/Ibope é conduzida anualmente, desde 1996, junto aos usuários do Cadê?, sob a orientação do IBOPE. (...) A quarta edição da pesquisa durou 5 semanas no período de Novembro a Dezembro de 1999 e **a análise dos resultados está baseada em cerca de 25 mil repostas.** (nossa ênfase)

Esses usuários forneceram suas respostas a um questionário de perguntas fechadas disponibilizado online pelo Cadê?. Seguem-se alguns exemplos dessas perguntas, que serão numerados para tornar a exposição mais fácil.

(1) Classifique seu interesse na Internet por cada um dos assuntos abaixo entre:

Muito Interessante Interessante Nenhum Interesse

Música 5 4 3 2 1

Notícias 5 4 3 2 1

Esportes 5 4 3 2 1

Turismo 5 4 3 2 1

Compras 5 4 3 2 1

Ciências 5 4 3 2 1

Artes 5 4 3 2 1

Sexualidade 5 4 3 2 1

Informática 5 4 3 2 1

Internet 5 4 3 2 1

¹ Más interpretações bastante comuns desse raciocínio geram dois problemas principais. O primeiro desses problemas é que muitos são levados a pensar que a solidez dos resultados está intimamente ligada ao tamanho da amostra. O segundo é a crença infundada de que os resultados das pesquisas quantitativas podem ser generalizados para populações diferentes daquela que deu origem aos resultados (essa crença tem, em sua base, uma concepção universalista da constituição do ser humano). Isso pode ser observado quando resultados de pesquisas realizadas em outros países são utilizados para orientar o que é feito no Brasil. Nesses casos, fatores culturais, sociais, econômicos, políticos, históricos, etc. são deixados de lado.

(2) Qual destas atividades é a mais importante para você na Rede?

Correio Eletrônico (e-mail)	Download de programas
Navegar na Web	Outros
Salas de Bate Papo (Chat)	

(3) O que mais mudou no seu dia a dia, desde que você passou a utilizar a Internet?

Assiste menos TV	Dorme menos
Lê menos revistas ou livros	Deixou de fazer outras atividades
Lê menos jornais	Nada mudou
Sai menos de casa	

Examinemos, agora, as principais características dessas perguntas. A primeira que salta aos olhos é que todas são perguntas fáceis de serem respondidas na medida em que exigem dos respondentes somente um click do mouse nos boxes correspondentes (que não foram reproduzidos acima). A segunda é que são perguntas para as quais as respostas já foram previstas pelos pesquisadores. Por isso as chamamos de fechadas. Nos três exemplos acima, somente o exemplo (2) permite respostas não previstas, respostas essas que não poderão ser analisadas separadamente, ou mesmo simplesmente conhecidas, pois farão parte de uma única categoria, a de "outros". A terceira característica decorre desta última: não há espaço para respostas espontâneas, críticas ou sugestões. A quarta, que também está relacionada às duas anteriores, fica mais evidente na pergunta (3), sobre as mudanças introduzidas pela Internet no cotidiano dos usuários. As mudanças previstas são mudanças de comportamento visíveis. Não há como fazer emergir aquilo que é invisível muitas vezes para o próprio usuário – como, por exemplo, as razões por trás dessas mudanças – a partir de respostas de múltipla escolha.

A análise dos dados coletados a partir desse tipo de questionário é baseada em raciocínios estatísticos e na geração de percentuais, médias e desvios-padrão. Na realidade, pelo menos no caso da pesquisa que estamos usando como exemplo, o principal objetivo é o de saber *qu岸tos* fazem, pensam, acham, etc. isso ou aquilo. Esses procedimentos dão sustentação a generalizações que apontam importantes tendências gerais, mas que pouco revelam a respeito de diferentes grupos de usuários. Seguem-se algumas delas.

As principais características de navegação desta comunidade são :

Seu caráter utilitário. As atividades mais importantes são o e-mail (39%) e a navegação (35%), dirigida sobretudo à busca por informações sobre produtos ou serviços (79%).

Seu caráter endógeno. A própria Web, seguida da informática, são os principais assuntos de interesse de quem navega (83% e 71%, respectivamente, consideram esses assuntos muito interessantes).

O conhecimento de que 83% dos usuários que forneceram suas respostas à pesquisa se interessam por assuntos da própria Web e 71% por assuntos relativos à informática, por exemplo, nada nos diz a respeito *do que* na Web desperta o interesse de usuários com diferentes perfis e, principalmente, *por que* o faz.

Isso não quer dizer que esses resultados sejam irrelevantes ou de pouco interesse. Eles nos dão uma importante visão panorâmica de usos e preferências, que abre nossos olhos para tendências e magnitudes de mudanças e pode servir de base para investigações mais aprofundadas. O que estamos tentando mostrar é que as pesquisas quantitativas produzem conhecimento sobre usuários que raramente correspondem aos homens, mulheres e crianças de carne-e-osso que usam computadores. Em outras palavras,

chamamos atenção para o fato de que os resultados das pesquisas quantitativas revelam as características gerais de usuários idealizados e abstraídos de contextos reais de existência por meio de procedimentos estatísticos.

Para identificar as necessidades, demandas e preferências de usuários reais são necessários um outro tipo de coleta de dados e um outro tipo de análise; uma coleta de dados e um tipo de análise que são bastante utilizados dentro da sub-área da psicologia que se dedica a tornar visíveis processos, motivações, desejos, preferências, conflitos, ansiedades, etc. que os próprios usuários desconhecem, ou seja, a da psicologia clínica.

3.2. A pesquisa qualitativa e a identificação das características de *um* perfil de usuário

As pesquisas qualitativas surgiram, na psicologia internacional e nacional, como uma reação à insensibilidade das pesquisas quantitativas na coleta de dados subjetivos. Por essa razão fundadora, a apresentação das diversas modalidades de pesquisa qualitativa é geralmente feita tomando como contraponto as pesquisas quantitativas².

No que se segue, não abriremos exceção a esse procedimento. Na realidade, ele nos guiará na apresentação de uma metodologia que poderá, em um primeiro momento, ser vista como pouco ortodoxa por um público mais sintonizado com os procedimentos metodológicos das ciências exatas. Esta apresentação contrapontual – cujo resumo acha-se disponível no quadro a seguir – será também pouco tradicional na medida em que freqüentemente interrompida por comentários que tentarão identificar como determinados procedimentos podem auxiliar o desenvolvimento de software e interfaces.

PESQUISA QUANTITATIVA	PESQUISA QUALITATIVA
Grandes amostras heterogêneas	Pequenas amostras homogêneas
Questionários de perguntas fechadas <i>Respostas prontas</i> <i>Quanto?</i>	Entrevistas de perguntas abertas Respostas livres Por quê? Como? Críticas e sugestões
-	Novos e inesperados pontos de vista
Análise estatística das respostas dadas a categorias pré-estabelecidas	Análise qualitativa das categorias recorrentes no discurso livre dos entrevistados
Visão panorâmica de uma multidão de usuários	Conhecimento aprofundado de um tipo de usuário

Podemos começar pelas diferenças constantes dos próprios títulos dados à seção anterior e a esta. Enquanto, na primeira, fizemos referência à identificação de *perfis* de usuários, nesta estamos nos referindo à identificação das características de *um perfil* de usuário. (Na realidade, podem ser investigadas, simultaneamente, as características de mais de um perfil, mas nunca de um número indeterminado de perfis).

Esta diferenciação pode se mostrar valiosa para o desenvolvimento de software, ao menos quando se tem perfis específicos em vista. Vejamos.

Enquanto na pesquisa do Cadê/Ibope, que estamos usando como exemplo, os dados foram coletados a partir de uma amostra grande e, por força, heterogênea, uma pesquisa qualitativa coleta seus dados a partir de pequenas amostras homogêneas de perfis

² É importante deixar claro, no entanto, que, como indicamos anteriormente, essas pesquisas podem ser usadas de forma complementar.

minuciosamente traçados (geralmente são usados de 20 a 100 sujeitos). A homogeneidade e a “alta definição” do(s) perfil(perfis) da amostra compensam seu pequeno tamanho. Nas grandes amostragens usadas pelas pesquisas quantitativas, há dezenas de diferentes perfis que, se tomados individualmente, têm relativamente poucos representantes dentro da amostra. (Na pesquisa qualitativa, o tamanho da amostra tem que ser pequeno porque, como veremos abaixo, a singularidade de cada entrevista é levada em conta na análise dos dados.)

Esses diferentes procedimentos podem ser melhor apreendidos a partir da inspeção dos primeiros blocos de perguntas – sobre dados de identificação dos respondentes – do questionário online do Ibope:

Nome	Você mora no:
E-mail	Brasil
Data de Nascimento	Exterior
Sexo	
	Se você mora no Brasil preencha os campos abaixo:
Estado Civil:	Cidade
Solteiro(a)	Estado
Casado(a)	CEP
Separado(a)	
Viúvo(a)	
FALE SOBRE VOCÊ	
Escolaridade:	Atualmente você:
Pós-Graduação Completa	Trabalha e estuda
Superior Completo/ Superior Incompleto	Trabalha
Segundo Grau Completo/ Segundo grau incompleto	Estuda
Primeiro Grau Completo/ Primeiro Grau Incompleto	Aposentou-se ou é pensionista
	Não é economicamente ativo
	Caso trabalhe, responda as próximas duas questões
Fala Inglês?	Atividade Principal:
Sim	Posição na Ocupação:
Não	
VOCÊ E A INTERNET	
Acessa a Internet: (marque todas as respostas que se aplicarem)	
do trabalho	da escola
de casa	outros
Usa a Internet há:	
Menos de 6 meses	de 1 a 2 anos
de 6 a 12 meses	Mais de 2 anos

É óbvio que essas perguntas se destinam a fornecer dados a partir dos quais serão compostos diferentes perfis de usuários e será identificado o perfil majoritário – aquele que chamamos de idealizado anteriormente –, cujas características serão forçosamente gerais. Esses são os dados que permitem aos pesquisadores a construção da seguinte “fisionomia da comunidade online”.

Concentrada no sudeste (57% é de SP, RJ ou MG), ainda há um forte traço masculino (os homens correspondem a 63% dos respondentes), embora a participação feminina tenha aumentado consideravelmente em relação a última pesquisa (Agosto 1998).

No auge de seu vigor e juventude (de 15 a 29 anos é o principal grupo etário com 68%), livre de compromissos sérios (79% solteiros ou separados), voltada à sua própria formação (69% estuda, 55% fala inglês) e sustento (64% trabalha), qualificada para o consumo (renda mensal familiar situada entre 10 e 50 salários mínimos para 59%).

A pesquisa qualitativa, em contrapartida, lida com usuários reais: homens, mulheres ou crianças situados em seus contextos sociais, econômicos, culturais, etc. de existência. Por essa razão, a pesquisa qualitativa tem início na seleção da fisionomia – ou perfil – cujas características queremos investigar. Ou seja, na pesquisa qualitativa, a fisionomia dos usuários é o ponto de partida e não o ponto de chegada da investigação. Recorrendo a um exemplo de fácil visualização, se quisermos saber o que na Web é do interesse de homens de classe média, profissionais liberais, solteiros, com idades entre 30 e 35 anos, não recrutaremos ninguém que não apresente esse perfil detalhado, que chamamos de “alta definição”.

Outra importante diferença entre as pesquisas quantitativas e qualitativas diz respeito a como são coletados os dados. Vimos que na pesquisa do Cadê/Ibope, que estamos usando como contraponto, o instrumento de coleta de dados era um questionário online de perguntas fechadas. Questionários como esse são comuns nas pesquisas quantitativas justamente porque permitem a quantificação das respostas.

Já na pesquisa qualitativa a coleta de dados é feita de um modo bem diferente. Nesta, fazemos uso de entrevistas de perguntas abertas (nas quais procuramos preservar ao máximo as características de descontração e informalidade de uma conversa cotidiana) baseadas em roteiros previamente construídos. A entrevista permite que entremos em contato individual com nossos sujeitos e as perguntas abertas permitem que eles se pronunciem livremente sobre os diversos tópicos que lhes são colocados. Trocando em miúdos, isso quer dizer que, uma vez feita uma pergunta (oralmente, em entrevistas face-a-face, ou por escrito, em entrevistas online), o entrevistado pode respondê-la da forma que bem entender. Assim é que, continuando com o exemplo dado acima, se quisermos saber o que na Web atrai o interesse de homens com um determinado perfil, faremos a vários homens que tenham esse perfil uma mesma e simples pergunta: “*O que na Web atrai seu interesse?*”.

Esses procedimentos dão ao pesquisador a oportunidade única de ouvir críticas e sugestões e de entrar em contato com novos pontos de vista que ele jamais poderia ter previsto. Além disso permitem que os pesquisadores aprofundem sua investigação por meio de simples perguntas como “Por quê?” ou “Como?”. As respostas dadas à pergunta-exemplo que apresentamos no parágrafo anterior seriam, deste modo, imediatamente seguidas de um “Por quê?”. É bom lembrar que perguntas simples e abertas não sugerem respostas (temos que pensar no problema da indução) e podem, caso haja espaço para isso como na metodologia que estamos apresentando, gerar todo tipo de respostas.

Tentando olhar para essas características a partir da ótica daqueles que desenvolvem software, acreditamos que o conhecimento das críticas e sugestões de usuários, de suas razões para preferir esse ou aquele software e/ou essa ou aquela interface, bem como de seus modos de uso desse ou daquele recurso, pode ser de grande utilidade. Nada impede, como acontece no caso das perguntas de respostas fechadas, que os usuários forneçam importantes pistas para a resolução dos problemas desses profissionais.

Obviamente o material discursivo gerado por esse tipo de metodologia não se presta a quantificações. Sua análise é conduzida a partir de várias leituras das transcrições, ou *logs*, das entrevistas em busca das categorias que emergem dos discursos dos entrevistados (a esse respeito, ver, por exemplo, Nicolaci-da-Costa, 1994). Essas categorias são detectadas a partir de sua recorrência nesses discursos. Também são analisadas as contradições e inconsistências presentes nesses discursos, contradições e inconsistências essas que são extremamente reveladoras de aspectos invisíveis do funcionamento humano como, por exemplo, motivações e desejos. Um exemplo desse tipo de análise pode ser encontrado no livro *Na malha da Rede: Os impactos íntimos da Internet* (Nicolaci-da-Costa, 1998), que foi integralmente escrito a partir das categorias que emergiram dos depoimentos de usuários da Internet nos primeiros tempos da difusão desta no Brasil.

Dado que a metodologia que vimos apresentando é pouco ortodoxa para quem está acostumado a procedimentos objetivos e exatos, sabemos que podem existir dúvidas a respeito da confiabilidade dos resultados por ela gerados. Acreditamos, no entanto, que a confiabilidade de um resultado não depende exclusivamente do que dizem os procedimentos estatísticos.

Claro está que os resultados gerados por esse tipo de metodologia não podem ser generalizados para populações que tenham um perfil de “alta definição” diferente do original. (Na realidade, nenhum resultado – mesmo quantitativo – pode ser generalizado para populações diferentes daquela na qual se originou.) Se o perfil investigado numa pesquisa qualitativa como a que estamos propondo foi o de homens de classe média, solteiros, profissionais liberais, com idades entre 30 e 35 anos, por exemplo, os resultados obtidos não poderão ser generalizados para, digamos, homens com idades entre 60 e 65 anos, mesmo que estes apresentem todas as outras características do perfil investigado.

A experiência de utilização dessa metodologia para diferentes finalidades ao longo de anos revela, no entanto, que esses resultados permitem generalizações bastante confiáveis para a população cujo perfil de “alta definição” foi investigado. Isso porque, quando se busca analisar porquês, como's, contradições e inconsistências gera-se um conhecimento profundo das raízes sociais de uma determinada configuração psicológica. E raízes sociais são comuns a homens, mulheres e crianças que têm um mesmo perfil.

4. Concluindo: breve nota sobre a implementação da metodologia proposta

Tornar visíveis processos e estados internos não é tarefa fácil. Por isso mesmo, os métodos de coleta e análise de dados usados em uma pesquisa qualitativa que tenha essa finalidade são trabalhosos e requerem um tipo de treinamento bastante distante do cotidiano das áreas tecnológicas. Essa constatação pode gerar uma dúvida a respeito de sua utilidade para os profissionais da informática: como poderão estes implementá-los?

Essa dúvida, no entanto, ocorrerá única e exclusivamente como resultado de um raciocínio disciplinar. Uma visão interdisciplinar da produção do conhecimento certamente dirá que essa é uma metodologia que requer a contribuição de profissionais com diferentes tipos de formação e treinamento. Diferentes competências podem – e, a nosso ver, devem – se complementar em torno de um objetivo comum. Esta foi a visão que norteou a apresentação deste trabalho.

Referências

- Birman, J. (1997). "Entre o gozo cibernético e a intensidade ainda possível: sobre Denise está chamando, de Hal Salwer", In: *Estilo e modernidade em psicanálise*. São Paulo, Editora 34, pp. 221-233.
- Breton, P. (1987). *História da Informática*. São Paulo, Editora da Universidade Estadual Paulista.
- Costa, A.C.A. (2001). IRC: uma nova alternativa para as relações entre as pessoas. Dissertação de Mestrado. Curso de Pós-Graduação em Psicologia Clínica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ.
- Dertouzos, M. (1997). *O que será: como o novo mundo da informação transformará nossas vidas*. São Paulo, Companhia das Letras.
- Gates, B. (1995). *A estrada do futuro*. São Paulo, Companhia das Letras.
- Leitão, C. e Nicolaci-da-Costa, A. M. (no prelo). "Psicologia clínica e informática: por que essa inusitada aproximação?", In: *Psicologia Clínica*.
- Nicolaci-da-Costa, A. M. (2000). "A tecnologia da Intimidade", In: *Anais do III Workshop de Fatores Humanos em Sistemas Computacionais*. Porto Alegre, Sociedade Brasileira de Computação, pp. 155-177.
- Nicolaci-da-Costa, A. M. (1998). *Na malha da Rede: Os impactos íntimos da Internet*. Rio de Janeiro, Editora Campus.
- Nicolaci-da-Costa, A. M. (1994). "A análise de discurso em questão", In: *Psicologia: Teoria e Pesquisa*, 10 (2), pp. 501-504.
- Romão-Dias, D. (2001). *Nossa plural realidade: um estudo sobre a subjetividade na era da Internet*. Dissertação de Mestrado. Curso de Pós-Graduação em Psicologia Clínica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ.
- Zaremba, R. (2001). *Escrevendo (ou seria 'teclando'?) o homem do século XXI*. Dissertação de Mestrado. Curso de Pós-Graduação em Psicologia Clínica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ.
- Zaremba, R.; Abreu, R.S. e Nicolaci-da-Costa, A.M. (2000). *A escrita digital: uma pedra no sapato da escola*. In: *Anais III Workshop sobre Fatores Humanos em Sistemas de Computação*. Porto Alegre, Sociedade Brasileira de Computação, pp. 196-202.

Interação com Agentes Pedagógicos Animados: Um Estudo Comparativo

Patrícia A. Jaques, Tânia Kist, Evandro Franzen, Marcelo Pimenta, Rosa Vicari

Programa de Pós-Graduação em Computação - Instituto de Informática – UFRGS
Bloco IV, Campus do Vale, Av. Bento Gonçalves 9500, Porto Alegre, RS, Brasil.
{pjaques, tania, mpimenta, rosa}@inf.ufrgs.br, evandro@open.inf.br

***Abstract:** This article presents a comparative study of the interaction aspects in animated pedagogical agents. Animated pedagogical agents are intelligent and autonomous agents that utilize learning strategies and multimedia resources to offer to the user an animated personage for interact and critic/comment/suggest actions carried out by the learner in the teach/learning process. This study permitted us to raise the necessary criteria to model and implement a pedagogical agent for mediate interaction in synchronous and collaborative tools for distance education.*

***Resumo:** Este artigo apresenta um estudo comparativo dos aspectos de interação em agentes pedagógicos animados. Agentes pedagógicos animados são agentes autônomos e inteligentes que utilizam estratégias de ensino e recursos multimídia visando oferecer ao usuário um personagem animado para interagir e criticar/comentar/sugerir ações realizadas no processo de ensino/aprendizagem. Este estudo nos permitiu levantar os critérios necessários para a modelagem e implementação de um agente pedagógico para mediar interação em ferramentas síncronas e colaborativas para ensino à distância.*

1. Introdução

Atualmente, muitos dos sistemas de ensino-aprendizagem utilizam a tecnologia de agentes na sua concepção. Nesta abordagem orientada a agentes, a arquitetura modular do sistema é substituída por uma sociedade de agentes que trabalham de forma cooperativa usando diversas técnicas de Inteligência Artificial (IA) e integrados a um Sistema Tutor Inteligente (STI). Segundo Vicari (1990), os STI são programas que, interagindo com o aluno, modificam sua base de conhecimento e possuem a capacidade de aprender e adaptar as estratégias de ensino de acordo com o modelo cognitivo construído a partir do diálogo com o aluno. O objetivo principal desses sistemas é proporcionar uma instrução adaptada ao aluno no conteúdo e na forma.

Um sistema de ensino que possui estratégias de ensino e formado por agentes é denominado de **Agente Pedagógico**. Nesses sistemas os agentes podem ser tanto usados para a representação de agentes personalizados e animados que representam um personagem que interage com o usuário, assim como agentes cooperativos que trabalham em *background* como parte da arquitetura do sistema educacional. O uso de agentes na concepção de sistemas educacionais traz algumas vantagens, tais como, reagir às ações do usuário, credibilidade, modelagem de sistemas colaborativos multi-usuário e modularidade e “openness”, pelo fato de que cada agente é um módulo único e independente do outro ficando mais fácil adicionar outros agentes a estes sistemas (Gürer, 1998).

Os agentes pedagógicos que utilizam a tecnologia de agentes sintéticos para a apresentação do material a ser estudado são conhecidos como **Agentes Pedagógicos Animados**. Os agentes sintéticos são agentes autônomos e inteligentes¹ que estão baseados na idéia de simulação de comportamento de seres vivos e inteligentes em máquinas. Para tanto, estes agentes utilizam recursos multimídia para oferecer ao usuário um personagem animado que lhe responda às ações realizadas.

A interface com o usuário é um componente importante, pois faz a mediação de toda a comunicação entre usuários e sistemas. Por definição, interface é uma zona de contato entre dois agentes e sua importância torna-se maior quando a diferença de natureza entre estes agentes é grande. Para sistemas educativos, a importância de uma interface de usuário de qualidade é ainda mais crucial, pois além de permitir contato com o aluno através do conteúdo (da mesma forma que o professor humano é mediador/facilitador do contato entre o aluno e o conhecimento), a interface deve continuamente estimulá-lo. Surpreendentemente, a profundidade com que é investigada não é proporcional à consciência de sua importância: em geral, os sistemas de ensino-aprendizagem se preocupam mais com teorias de aprendizagem, representação do conhecimento do que com novas técnicas de interação e/ou usabilidade de suas interfaces. Geralmente, apesar de visualmente ricas e com aparência e experiências impressionantes, permitem diálogos sem flexibilidade, com poucas modalidades de comunicação, pobre *feedback* e pouca usabilidade.

Nesse artigo são comparados os aspectos de interação de alguns agentes pedagógicos animados. Na seção 2 é realizada uma breve apresentação dos agentes pedagógicos animados: Adele, Steve, Vincent, Cosmo e Herman. Na seção 3 é apresentada uma tabela comparativa destes agentes em relação a alguns critérios de interação escolhidos. Na seção 4 é apresentada a proposta de um agente pedagógico animado para monitorar interação entre alunos em sistemas colaborativos de ensino. Essa proposta foi baseada nos requisitos levantados a partir do estudo realizado neste artigo. Finalmente, na seção 5 é apresentada a conclusão deste artigo, onde são colocadas as propostas de trabalhos futuros.

2. Análise da Interface de Agentes Pedagógicos Animados

Os agentes pedagógicos animados utilizam-se de recursos multimídia para proporcionar ao usuário um personagem animado com características semelhantes às de seres vivos e inteligentes. Estas características, tais como, expressões faciais e entendimento de emoções humanas, associadas a uma boa interface de diálogo com o usuário, irão tornar os agentes mais atrativos aos alunos pelo fato de explorarem modos de interação mais naturais (Elliot e Brzezinski, 1998). Desta maneira, diferentemente dos sistemas convencionais, a comunicação dos agentes pedagógicos animados possui uma natureza mais antropomórfica e social. A seguir apresentaremos as características principais de alguns agentes pedagógicos animados disponíveis na literatura.

2.1 Adele

A Adele² (Agent for Distance Education – Light Edition) é um agente pedagógico com expressões humanas que foi projetado para trabalhar com estudantes através de resolução

¹ Ver detalhes sobre agentes inteligentes em (Shoam, 1997) (Jennings et al., 1998) (Wooldridge e Jennings, 1997).

² <http://www.isi.edu/isd/ADE/ade-body.html>

de problemas associada à disponibilização de material para estudo (Jonhson et al., 1998) (Johnson et al, 2000) (Johnson e Shaw, 1997). A Adele foi desenvolvida no Centro para Pesquisa Avançada em Tecnologia para Educação da Information Science Institute da USC (Califórnia – EUA).

No caso de uma aplicação para diagnóstico clínico são apresentados aos estudantes materiais sobre um caso específico e, então, são fornecidos vários casos aos quais os estudantes devem trabalhar. Adele executa funções como destacar aspectos interessantes do caso, monitorar e fornecer *feedback* ao aluno de como ele pode trabalhar em um caso, dar sugestões para ações ou realizar um exame para verificar o grau de compreensão do aluno em relação aos princípios associados ao caso.

A arquitetura do sistema Adele consiste de três componentes principais: *interface gráfica de simulação (GUI)*, *máquina de raciocínio* (responsável pela tomada de decisão) e um *agente animado persona* (representação gráfica do agente).

Na Figura 1a podemos observar a interface gráfica de simulação para o caso de diagnóstico clínico e na Figura 1b podemos observar Adele explicando a um aluno a importância de apalpar o abdômen do paciente.



Figura 1a: Interface Gráfica de Simulação de Adele.

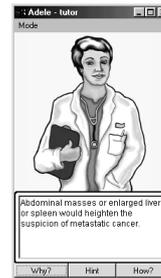


Figura 1b: Agente Adele

Adele é um personagem 2-D, representado apenas da cintura para cima (busto) e encontra-se fixa a uma janela, o que se caracteriza em uma considerável limitação. Por exemplo, no caso de diagnóstico clínico, o estudante de medicina deve fazer algumas simulações de exames médicos. Quando o aluno não sabe como realizar uma determinada tarefa, ele pede auxílio a Adele que expõe uma explicação oral e por escrito em uma janela de texto que aparece na mesma janela do personagem animado. Como a Adele se encontra em um sistema baseado em simulação seria mais interessante se a personagem pudesse realizar as ações na própria interface gráfica.

A Adele observa as ações do usuário e se comunica com ele usando um sintetizador de voz. O personagem interage com o usuário quando deve lhe apresentar algum exercício ou quando ele clica em um dos botões da Interface de Adele que podem ser: “How” (como realizar a ação sugerida por Adele); “Why” (porque realizar esta ação) e “Hint” (qual a próxima ação que deve ser realizada).

2.2 Steve

Steve³ (Soar Training Expert for Virtual Environment) é um agente pedagógico autônomo e animado inserido em um Sistema Tutor Inteligente construído para auxiliar alunos em cursos de treinamento navais. O treinamento acontece em um espaço tridimensional e interativo em Realidade Virtual (Rickel e Johnson, 1998).

Steve é representado por um personagem com habilidades de gesticulação, podendo demonstrar procedimentos através de gestos e comunicação verbal, tendo como funções principais acompanhar e monitorar o aprendizado do aluno durante a interação com o STI. Nas Figura 2a e 2b pode ser observada a interface do agente Steve.

A comunicação entre *Steve* e os alunos é feita em linguagem natural. Utilizando linguagem sintetizada ele reconhece pedidos de ajuda neste formato, sendo o processo de síntese e reconhecimento de linguagem realizado por um *software* especial, desenvolvido pelo *Entropic Research*.

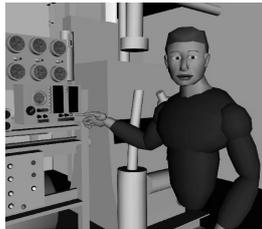


Figura 2a: O Personagem Steve.

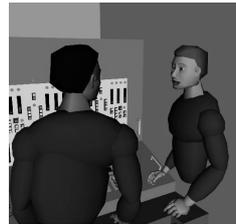


Figura 2b: Módulo de apoio a atividades em grupo.

Uma comunicação não-verbal também é aplicada na interação entre *Steve* e o aluno através dos equipamentos de imersão utilizados para navegar no ambiente, tais como luvas e capacetes, que capturam as informações necessárias à atividade de monitoramento de *Steve*. A partir das informações resgatadas dos equipamentos de imersão, *Steve* reconhece o campo de visão do aluno, podendo verificar se o aluno está olhando para o lugar certo, no mundo virtual, para executar os próximos passos da atividade. Ele pode mover a cabeça e olhos para localizar objetos no ambiente, dirigir a atenção para tais objetos, e olhar para o aluno enquanto fala ou espera uma ação. Estes movimentos de cabeça dão a impressão de que *Steve* está atento ao ambiente, bem como comprometido com a atividade do aluno.

2.3 Vincent

Vincent é um agente pedagógico animado para sistemas de ensino-aprendizagem baseados na *Web* (Paiva e Machado, 1998) (Paiva e Machado, 1999) (Paiva et al., 1999). Ele é um agente intrinsecamente autônomo porque age sem a interferência humana ou de outros agentes e não depende do ambiente em que está embutido, possuindo características de reusabilidade.

Vincent foi aplicado em um sistema de treinamento chamado TEMAI, fornecido pelo CTC (*Technological Shoe-Making Center*) e disponibilizado pelo LTC (*Local Training Center*),

³ <http://www.isi.edu/isd/VET/vet-body.html>.

um centro tecnológico europeu responsável por hospedar cursos de treinamentos à distância. Ele possui servidores WWW e MTSs (*Modular Training System*), sendo estes últimos responsáveis pelos serviços de gerenciamento dos cursos de treinamento e de *software* que rodam nos computadores dos alunos. No TEMAI, o agente *Vincent* pode se comunicar com os ambientes de aprendizado para acompanhar o aluno e atualizar o seu modelo.

Vincent é representado por um personagem que possui um conjunto de comportamentos apresentados através de recursos áudio-visuais que correspondem a atitudes emocionais. Tais atitudes são alteradas de acordo com a interação com o aluno e dependem de fatores, tais como, o modelo do aluno e seu comportamento. A Figura 3 apresenta algumas atitudes emocionais de *Vincent* e as respectivas situações em que são invocadas.



Figura 3: Atitudes Emocionais de Vincent.

Vincent possui dois tipos de comportamento, definidos em função de seus objetivos: cognitivo e reativo. O comportamento cognitivo tem por objetivo decidir qual ação pedagógica deve ser escolhida para uma situação particular e o comportamento reativo é responsável pelas atitudes áudio-visuais de *Vincent*.

2.4 Cosmo

Desenvolvido pelo Multimedia Laboratory Department of Computer Science North Carolina State University (NSCU), Cosmo⁴ é um agente que habita um ambiente de aprendizado sobre o roteamento de pacotes no domínio da Internet. Sua função é, em tempo real, demonstrar e aconselhar estudantes na melhor maneira de enviar pacotes para um determinado destino, através de um mundo virtual de roteadores conectados (Lester e Stone, 1996) (Lester et al., 2000) (Lester et al., 1996a) (Lester et al., 1996b). A Figura 4 apresenta o agente Cosmo e o ambiente Internet Advisor.

Cosmo tem a aparência de uma criatura estranha com antenas, muito semelhante a um pequeno robô humanóide. A utilização de um ponto de interrogação no seu corpo é bastante significativa e indica imediatamente que se trata de uma forma de auxílio ao usuário. A cabeça e as mãos grandes e desproporcionais ao corpo indicam que estes componentes do agente são os principais responsáveis pela interação com o ambiente. É através das mãos que o agente aponta para algum componente da interface e de acordo com a direção que o agente olha é possível definir o foco de ação.

⁴ IntelliMedia Project – Universidade da Carolina do Norte - <http://www.csc.ncsu.edu/eos/users/l/lester/www/iimedia/index.html>



Figura 4: Cosmo e o Ambiente Internet Advisor

2.5 Herman, o Inseto

Herman⁵ é um agente pedagógico que habita um ambiente de aprendizado baseado em domínios de botânica e fisiologia. Sua função é auxiliar estudantes a resolver problemas como o desenvolvimento de uma planta ou montagem correta das várias partes que a compõe. Dado um conjunto de condições ambientais, crianças podem graficamente montar plantas que podem sobreviver em ambientes específicos (Lester e Stone, 1997) (Tows et al., 1998) (Elliot et al., 1997) (Lester et al., 1997). A Figura 5 ilustra um ambiente específico para o qual um estudante deve montar uma planta.

Herman é um falador e ágil inseto com propensão a voar sobre a tela e com a função de dirigir-se aos aspectos e componentes para montagem de plantas. No processo de explanação de conteúdos, ele executa uma ampla faixa de atividades que incluem caminhar, voar, encolher-se, expandir-se, deitar e outras.



Figura 5: Agente Herman aguardando o usuário realizar uma ação.

As atividades são organizadas por um componente de seqüenciamento de atividades existente na arquitetura do agente. O seqüenciamento das atividades é dinamicamente definido e baseado em princípios de concorrência entre as várias atividades existentes no espaço de atividades.

3. Estudo Comparativo

A seguir é apresentada uma tabela comparativa em relação aos critérios considerados relevantes para um estudo dos aspectos de interação de agentes pedagógicos animados (Ver Tabela 1). Esses critérios foram classificados em 3 tipos principais: técnicos, contextuais e

⁵ IntelliMedia Project – Universidade da Carolina do Norte - <http://www.csc.ncsu.edu/eos/users/l/lester/www/iimedia/index.html>

de interação. Os critérios técnicos dizem respeito às tecnologias utilizadas e necessárias no desenvolvimento e à execução do sistema. Os critérios contextuais informam o contexto em que o agente funciona. Por exemplo, o ambiente em que ele se encontra. Por último, os critérios de interação buscam identificar os meios pelos quais se dá a comunicação entre o agente e o aluno.

A tabela 1 mostra uma comparação dos Agentes Pedagógicos Animados apresentados na seção 2. O preenchimento da tabela segue a seguinte convenção: cada célula ou possui uma descrição de como o agente atende este critério (por exemplo, "inseto" para o critério "personagem"), ou o símbolo (??) para indicar simplificada e a dúvida dos autores em relação ao posicionamento de um agente em relação a este aspecto, a partir das informações disponíveis.

Tabela 1: Comparação entre Agentes Pedagógicos Animados

	CRITÉRIOS	ADELE	STEVE	VINCENT	COSMO	HERMAN
C O N T E X T O	Domínio	Medicina	Treinamento Naval	Não depende de domínio	Roteamento de pacotes na Internet	Ensino de Botânica e Fisiologia.
	Contexto de Uso	Prontuário Médico	Sala de Máquinas de um Navio	Variado	Rede de Roteadores Conectados	Micromundo formado por plantas
T É C N I C O	Linguagem de Implementação	applet Java	SOAR Tcl/Tk e C, VRML	??	C++ e MS Game Developer's Kit	C++ e MS Game Developer's Kit
	Plataforma De Suporte	-Agente: Qualquer plataforma -Sintetizador de voz: Windows	Ambiente Virtual de Treinamento VIVIDS	Servidores WWW e Sistema de Treinamento Modular (MTS)	Windows e Framework para agentes animados em 3D	Windows e Framework para agentes animados em 3D
I N T E R A Ç Ã O	Papel do Personagem	Médica	Homem	Homem	Robô	Inseto
	Aparência	2-D, Busto	3-D, Busto	2-D, Corpo	3-D, Busto	3-D, Corpo
	Comunicação com o Usuário	E: Teclado S: Voz, Texto (vídeo)	E: Ling. natural Ações do usuário no ambiente S: Voz, Ações do agente no ambiente	E: Ações do usuário no ambiente S: Voz, Texto	E: Ações do usuário no ambiente S: Voz, Ações do agente no ambiente	E: Ações do usuário no ambiente S: Voz, Ações do agente no ambiente
	Ações	Responde a -Hints -How -Why	Acompanha aluno. Simulações Explicações	Representa atitudes audiovisuais animadas. Explicações.	Explicações. Locomove-se e aponta objetos.	Explicações. Locomove-se e aponta objetos.
	Expressões Faciais	Poucas e limitadas	Olha para o ambiente e aluno	Significativo conjunto de expressões faciais.	Significativo conjunto de expressões faciais	Significativo conjunto de expressões faciais.
	Comportamento o Cognitivo	Modelo do aluno. Modelo de Tarefas. Tomada de Decisão.	Modelo do aluno. Planejador de Tarefas. Tomada de Decisão. Interpretação em ling. natural.	Modelo do aluno. Modelo de Tarefas. Tomada de Decisão.	Modelo do aluno. Modelo do Mundo. Planejador de comportamentos. Tomada de decisão.	Modelo do aluno. Framework de seqüenciamento de comportamentos. Tomada de decisão.
	Credibilidade	Comportamento estático e previsível. Poucas expressões faciais emocionais.	Ótima consciência corporal. Poucas expressões faciais emocionais.	Apresenta conjunto variado de atitudes emocionais.	Apresenta reações emocionais limitadas.	Apresenta reações emocionais limitadas.

A linguagem de implementação utilizada determina a portabilidade, performance, bem como recursos audiovisuais que podem ser utilizados no sistema. Dentre os agentes

citados, o único portátil é a Adele que foi desenvolvida em Java⁶ (Ver critérios técnicos na Tabela 1) e que, portanto, pode ser executada em qualquer navegador World Wide Web (browser) ou sistema operacional que possua suporte à Java. O Steve foi desenvolvido utilizando as linguagens SOAR, Tcl/Tk, C e VRML e é executado no ambiente virtual VIVIDS. Os outros sistemas foram desenvolvidos para plataforma Windows, pois foram implementados em C++ com MS Game Developer's Kit.

Uma das principais questões a serem resolvidas na utilização de agentes que possuem aparência de seres vivos e que habitam ambientes de simulação é que estes, assim como humanos, devem ser capazes de referir e interagir com objetos existentes em seu ambiente através da combinação de fala, locomoção e gestos. Agentes animados devem ser capazes de mover através de seu ambiente, apontar e referir objetos apropriadamente para prover auxílio na resolução de problemas de acordo com o contexto. Estes aspectos foram analisados nos critérios do tipo **interação** (Ver Tabela 1).

Em relação à personagem, Adele tem a aparência de uma médica, Steve e Vincent de um rapaz, Cosmo de um robô e Herman de um inseto. Todos os personagens possuem uma identidade visual condizente ao ambiente de aprendizagem em que estão inseridos (Ver critério **Papel do Personagem** na Tabela 1).

Vincent e Herman são representados de corpo inteiro, enquanto que Adele, Steve e Cosmo possuem apenas o busto representado (corpo da cintura para cima). A representação em busto, nestes casos, não é necessária nas atividades realizadas por estes agentes, porém, esta representação dos agentes não está fidedigna a representação do mundo real (Ver critério **Aparência** na Tabela 1).

Um outro aspecto essencial na interação entre agente pedagógico e aluno é a forma de comunicação utilizada entre estes dois personagens (Ver critério **Comunicação com o Usuário** na Tabela 1). Para os personagens animados, a conversação em linguagem natural exerce um papel importante, pois é semelhante à linguagem utilizada pelas pessoas no dia-a-dia. O único agente que reconhece entrada em linguagem natural é o Steve. A interação de Adele com o usuário se dá de maneira convencional, onde o aluno pode escolher uma das opções de botões pré-definidas. Cosmo e Herman captam as ações dos usuários. Steve responde ao usuário através de ações no ambiente e em linguagem natural falada (voz), assim como, Adele, Vincent, Herman e Cosmo. Além disso, Vincent é capaz de realizar ações no ambiente, tais como, demonstrar simulações ao usuário, apontar, entre outras, e responde as ações do usuário através de texto. Adele, além de responder verbalmente, gera resposta em texto.

Steve é o agente que apresenta maior riqueza de comportamentos (Ver critério **Ações** na Tabela 1). Por exemplo, Steve apresenta demonstrações e explicações de simulações a serem ensinadas e corrige ações errôneas dos usuários com demonstrações. Steve é capaz de se locomover, apontar para objetos e olhar em direção ao usuário. Vincent representa um conjunto de atitudes audiovisuais que exprimem o seu sentimento em relação ao aluno. Cosmo se locomove em direção aos objetos do sistema e realiza demonstrações ao aluno. Tanto Vincent, Cosmo e Herman realizam explicações ao usuário quando verificam que ele realizou uma ação não correta ou convidam o usuário à execução de alguma atividade.

⁶ Java é marca registrada de Sun Microsystems, Inc. <http://java.sun.com>

Adele responde ao usuário apenas com três opções de perguntas: Sugestões (Hint), Como (How), Por quê (Why).

Uma outra questão importante na análise de interfaces de agentes animados são as **expressões faciais**, pois elas dão maior realismo ao personagem, além de proporcionar um outro tipo de feedback às ações do usuário, tais como, reprovação, aprovação e espera (El-Nasr et al, 1999) (Ver critério Expressões Faciais na Tabela 1). Adele possui um repertório de expressões faciais e corporais que representam emoções, tais como, surpresa e desapontamento. Esse repertório, porém, é bastante restrito e um pouco estático, perdendo o seu realismo. Steve possui poucas expressões faciais. Sua riqueza maior está nos movimentos da cabeça. Vincent apresenta um conjunto considerável de expressões faciais que estão integradas aos seus comportamentos audiovisuais. Cosmo também apresenta um conjunto significativo expressões faciais e Herman apresenta poucas expressões.

Um outro critério avaliado é a presença de **comportamento cognitivo** nos agentes (Ver critério Comportamento Cognitivo na Tabela 1). Os agentes pedagógicos devem oferecer um estudo individualizado ao aluno e, para tanto, todos os agentes possuem, mantêm e atualizam os dados dos alunos em um modelo de aluno que guarda informações sobre o nível de conhecimento e as preferências deste aluno. Os agentes usam as informações do modelo do aluno, para definir qual o melhor conteúdo e/ou exercício a ser exibido para o aluno. As informações do modelo do aluno também permitem ao agente possuir um outro importante comportamento cognitivo que é a tomada de decisão, ou seja, a cada interação com o aluno, o agente define qual a ação que deve ser executada por ele.

O Steve além dessas tarefas cognitivas é ainda capaz de interpretar as mensagens em linguagem natural do aluno.

Um outro importante mecanismo cognitivo de agentes pedagógicos é o modelo de tarefas, onde cada tarefa é descrita em função das suas metas, ações a serem executadas, ordem de execução destas ações e efeitos destas em relação às demais. O modelo de tarefas permite ao agente saber quando o aluno está realizando uma ação errada e intervir lhe dando uma melhor explicação. Steve apresenta um modelo de tarefas das ações que devem ser executadas por eles e sua ordem.

Adele armazena o seu plano de ação em um mecanismo de sua arquitetura chamado *Planejador de Tarefas*. O Planejador de Tarefas de Adele não é um mecanismo tão sofisticado quanto o modelo de tarefas de Steve, que mantém também relação entre as tarefas, já que as tarefas realizadas por Steve são mais complexas, pois são realizadas em um ambiente virtual, onde muitas variáveis devem ser mantidas.

Vincent possui um *módulo corpo* que é responsável por realizar as tarefas pedagógicas e de diagnóstico. Herman mantém o plano de ações em um *mecanismo de seqüenciamento de comportamentos*: um mecanismo que mantém a ordem de realização das ações. O agente Cosmo possui um componente chamado *Agent Behaviour Planner* que é usado, juntamente com um modelo do mundo (do ambiente de simulação), para planejar e decidir suas ações. Este componente opera segundo o *framework de seqüenciamento de comportamentos* de Herman.

Um aspecto de interação que é muito importante em agentes animados é a **credibilidade** (Ver critério Credibilidade na Tabela 1). Segundo Loyall e Bates (Loyal e Bates, 1997) credibilidade é usado no senso de atores credíveis na arte dramática, significando que o

público ou usuário pode esquecer de seu ceticismo e sentir que o personagem ou agente é real. Tornar um agente credível envolve os aspectos necessários para o agente expressar sua personalidade. Existem ainda algumas ações que tornam o agente mais real como a movimentação dos olhos, tempo de pausa para falar, consciência da posição do corpo e espaço pessoal e comunicação em linguagem natural. Além dessas ações, um agente credível deve produzir emoções devido ao sucesso ou falha na comunicação como também em outras ações.

Quanto a esses aspectos de realismo, Adele se apresenta muito estática em uma janela e seu comportamento é bastante previsível, possuindo poucas expressões faciais e demonstrando reações emocionais pobres. Steve, por sua vez, apresenta ótima consciência corporal, porém, não apresenta expressões faciais emocionais. Vincent apresenta um repertório variado de expressões e comportamentos emocionais para serem usados em várias situações diferentes de interação com o usuário, porém essas atitudes são repetitivas, o que tira o realismo do personagem. Herman possui um número de reações faciais menor do que Cosmo. Cosmo tem uma maior capacidade de responder emocionalmente e tem maior número de comportamentos.

4. O Agente Pedagógico Proposto

A análise realizada neste artigo teve como objetivo auxiliar na especificação de critérios e funcionalidades necessários para a construção de um agente pedagógico animado que promova a interação social. Este agente está sendo modelado como parte da arquitetura multiagente do projeto “Um Modelo Computacional de Aprendizagem à Distância Baseada na Concepção Socio-Interacionista” (Vygotsky, 1998) (Levy, 1999) (Freire, 1995), ora em desenvolvimento, onde se encontram em andamento duas teses de doutorado e uma dissertação de mestrado (Andrade et al., 2001).

Esse sistema proposto é formado por cinco tipos de agentes artificiais – o agente de diagnóstico, o agente mediador, o agente colaborativo, o agente social e o agente semiótico – e os agentes humanos (aprendizes). O agente mediador é um agente de interface que é responsável por apresentar o conteúdo ao aluno. Toda as ações do usuário serão captadas pelo agente mediador e repassadas ao agente de diagnóstico. O agente de diagnóstico atualiza as informações no modelo do aluno e verifica, de acordo com os dados recebidos, se há a necessidade de utilizar uma nova tática de ensino e a envia ao agente mediador. Se essa tática for, por exemplo, a apresentação de um material instrucional, o agente mediador faz a solicitação deste material ao agente semiótico. O agente semiótico busca o signo ou instrumento solicitado em sua base de dados (por ex.: conteúdos, animações, vídeos, chat) e os envia ao agente mediador para ser exibido ao aluno. Quando o agente mediador verificar uma deficiência no aprendizado do aluno que seja interessante a atividade em grupo, ele fará a solicitação ao agente social. O agente social formará um grupo de alunos para estudo e criará um agente colaborativo para mediar e monitorar a interação entre estes alunos. Existe um agente de diagnóstico e um agente mediador para cada aluno, um agente social e um agente semiótico para toda a sociedade e um agente colaborativo para cada grupo de alunos formado.

Nesta arquitetura dois agentes serão implementados como agentes animados: o agente mediador e o agente colaborativo. O agente que discutiremos neste artigo é o agente colaborativo. Esse agente de interação social tem por função promover e mediar a interação

entre grupos de alunos em ferramentas síncronas de comunicação (por exemplo, chat). Para tanto, ele assiste os alunos durante as interações, incentivando-os quando se mostrarem desmotivados, apresentando novos conceitos e corrigindo concepções errôneas.

Devido a sua função social - comunicar-se com o usuário e promover e monitorar a interação entre alunos - é interessante que esse agente possua uma interface que permita explorar a natureza social do homem. Por isso, optamos por representá-lo como um agente animado que possui uma identidade e que interage com o aluno através de linguagem natural. Uma de nossas preocupações é explorar mais profundamente toda a potencialidade social do ser humano na aprendizagem. Estudos demonstram que pessoas interagindo com personagens animados aprendem a interagir com outros seres humanos (Huard, 1998).

Assim, como nas interações sociais humanas, o agente colaborativo deve perceber e exibir emoções. A **aprendizagem** é um processo maior onde não se encontra apenas a transmissão e o aprendizado de conteúdo. O tutor, no caso o agente colaborativo, deve promover o desenvolvimento emocional e afetivo do aluno, gerando-lhe autoconfiança e um estado de espírito positivo, mais ideais à aprendizagem. A forma como as perturbações emocionais interferem na vida mental não é novidade. Alunos mal-humorados, ansiosos ou deprimidos encontram maior dificuldade em aprender (Goleman, 1995).

Porém, para tanto, o agente tem de interpretar corretamente as emoções do aluno para lhe responder de maneira adequada. Por exemplo, digamos que o aluno tenha dificuldade de realizar seus exercícios, pois se encontra muito ansioso. Se o agente interpretar erroneamente o estado de espírito do aluno, ele poderá gerar uma reação que deixará o aluno mais ansioso e prejudicá-lo no seu estudo, ao invés de ajudá-lo. Para tanto, é necessário que ele possua não apenas um modelo cognitivo do aluno, mas também um modelo emocional. Na arquitetura proposta utilizaremos o modelo definido por Bercht (Bercht et al., 1999).

Temos que ter em mente a responsabilidade da utilização de uma arquitetura emocional para agentes interagirem com o usuário, principalmente na educação. Muitas vezes, observamos que os agentes possuem atitudes que não são apropriadas para o estado de espírito do aluno. Por exemplo, Vincent possui a atitude triste que é apresentada quando o aluno não conseguiu realizar um exercício. Esse tipo de atitude pode gerar uma reação perturbadora no aluno, o deixando mais ansioso e menos autoconfiante. É necessário que busquemos identificar quais os comportamentos são apropriados para promoverem um estado de espírito no aluno que lhe proporcione um melhor aprendizado.

Para conseguir esse objetivo, o agente também deve obter a empatia do aluno. Por exemplo, o aluno deve gostar do agente, se identificar com ele, confiar nele. Se o aluno não gostar do agente, ele não irá considerar relevantes as colocações do agente, podendo até evitar a utilização do sistema ou desabilitar a presença do agente.

Além disso, para que o aluno possua empatia pelo agente é necessário que o agente seja credível, ou seja, o aluno se envolve de tal maneira com o agente que acredita que ele seja real. Para tanto, o agente deve possuir um repertório de comportamentos variados e responder de maneira adequada e dinâmica às ações do aluno.

5. Conclusão

O estudo apresentado neste artigo foi resultado da avaliação da interação tutor e aluno em alguns agentes pedagógicos animados existentes. Esse estudo nos permitiu levantar requisitos e funcionalidades necessários para a definição de aspectos de interface e interação de um sistema de ensino voltado a aprendizagem colaborativa. Resumidamente, observamos a necessidade de explorar comportamentos emocionais nos agentes a fim de (a) oferecer uma interface mais natural ao aluno que considere a sua natureza social e (b) trabalhar aspectos afetivos de forma a proporcionar resultados positivos na aprendizagem (Goleman, 1995) (Hayes-Roth, 1998).

Inicialmente, iremos realizar os levantamentos que o estudo deste artigo apontou ser necessário: definição de comportamentos afetivos que propiciem um melhor aprendizado ao aluno, desenvolvimento de uma arquitetura emocional do agente para que responda emocionalmente de forma coerente às ações do aluno e integração da arquitetura do agente com o modelo do aluno proposto por Bercht (Bercht et al., 1999). Após essa etapa, será realizada a implementação em Java desse agente. No presente momento, o agente semiótico, o agente de diagnóstico e o agente mediador encontram-se em desenvolvimento.

6. Referências Bibliográficas

- (Andrade et al., 2001) ANDRADE, Adja; JAQUES, Patricia; VICARI, Rosa; BORDINI, Rafael; JUNG, João. A Computational Model of Distance Learning Based on Vygotsky's Socio-Cultural Approach. MABLE Workshop (AI-ED 2001) San Antonio, Texas, Maio 19-23 2001.
- (Bates, 1994) BATES, Joseph. The Role of Emotion in Believable Agents. *Communicat. of ACM*. Julho, 1994.
- (Bercht et al., 1999) BERCHT, M.; MOISSA, H.E.M; VICCARI, R.M. Identificação de fatores motivacionais e afetivos em um ambiente de ensino e aprendizagem. SBIE'99. Novembro 23-25, 1999, Curitiba, PR (Poster).
- (Elliot et al., 1997) ELLIOTT, Clark, RICKEL, Jeff e LESTER, James C. Integrating Affective Computing into Animated Tutoring Agents. Working Notes of the IJCAI '97 Workshop on Animated Interface Agents: Making Them Intelligent, pp. 113-121, Nagoya, Japan, August 1997.
- (Elliot e Brzezinski, 1998) ELLIOTT, Clark; BRZEZINSKI, Jack. Autonomous Agents as Synthetic Characters. *Software Agents*. BRADSHAW, Jeffrey (Ed.). AAAI Press/The MIT Press, 1998.
- (Elliot et al., 1999) ELLIOT, Clark; RICKEL, Jeff; LESTER, James. Lifelike Pedagogical Agents and Affective Computing: An Exploratory Synthesis. In M. Wooldridge and M. Veloso (Eds.), *Artificial Intelligence Today*, Lecture Notes in Computer Science 1600, Springer-Verlag, pp. 195-212, 1999.
- (El-Nasar et al., 1999) EL-NASAR, Magy; IOERGER, Thomas; YEN, John; HOUSE, Donald; PARKE, Frederic. Emotionally Expressive Agents. *Proceedings of Computer Animation '99*, 1999.
- (Freire, 1995) Freire, Paulo e Fagundes, Antônio. *Por uma Pedagogia da Pergunta*. Ed. Paz e Terra, Rio de Janeiro, 1995.
- (Goleman, 1995) GOLEMAN, Daniel. *Inteligência Emocional*. Objetiva, 1995.
- (Gürer, 1998) GÜRER, Denise. The Use of Distributed Agents in Intelligent Tutoring. In: ITS'98 Workshop On Pedagogical Agents, 4, 1998, San Antonio. p.20-25.
- (Hayes-Roth, 1998) HAYES-ROTH, Barbara. *Interacting with Animated Characters: Puppets, Bartenders and Auto Salespersons*. Stanford, 1998. Technical Report (KSL-98-07) – Stanford University.
- (Huard, 1998) HUARD, R. *Character Mastery with the Improvisational Puppets Program*. Stanford, 1998. Technical Report (KSL-98-11) – Stanford University.
- (Jennings et al., 1998) JENNINGS, N.; SYCARA, K.; WOOLDRIDGE, M. A Roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1:275-306, 1998.
- (Johnson et al., 1998) JOHNSON, Lewis; SHAW, Erin; GANESHAN, Rajaram. Pedagogical Agents on the Web. In: ITS'98 Workshop On Pedagogical Agents, 4, 1998, San Antonio. p.2-7.

- (Johnson e Shaw, 1997) JOHNSON, W. Leurs; SHAW, Erin. Using Agents to Overcome Deficiencies in Web-Based Courseware. World Conference On Artificial Intelligence In Education, 8, 1997, Kobe.
- (Johnson et al, 2000) JOHNSON, Lewis; RICKEL, Jeff; LESTER, James. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. URL: <http://www.isi.edu/isd/VET/apa.ps>
- (Lester et al., 1996a) LESTER, James C.; VOERMAN, Jennifer L.; TOWNS, Stuart G.; CALLAWAY, Charles B. Cosmo: A Life-like Animated Pedagogical Agent with Deictic Believability. URL: <http://www.csc.ncsu.edu/eos/users/l/lester/www/imedia/papers.html>.
- (Lester et al., 1996b) LESTER, James C.; CALLAWAY, Charles B; TOWNS, Stuart G. Creating Lifelike Behaviors in Animated Pedagogical Agents. In: ITS'98 Workshop On Pedagogical Agents, 4, 1998, San Antonio. **Anais...** San Antonio: 1998. p.75-77.
- (Lester et al., 1997) LESTER, James; CONVERSE, Sharolyn; KAHLER, Susan; BARLOW, Todd; STONE, Brian; BHOGAL, Ravinder. The Persona Effect: Affective Impact of Animated Pedagogical Agents. Proceedings of CHI '97, pp. 359-366, Atlanta, March 1997.
- (Lester e Stone, 1997) LESTER, James; STONE, Brian. Increasing Believability in Animated Pedagogical Agents. Proceedings of the First International Conference on Autonomous Agents, pp. 16-21, Marina del Rey, California, February 1997.
- (Lester et al., 2000) LESTER, James C., MORENO, Roxana e MAYER, Richard E. Life-Like Pedagogical Agents in Constructivist Multimedia Environments: Cognitive Consequences of their Interaction. URL: <http://www.csc.ncsu.edu/eos/users/l/lester/www/imedia/papers.html>.
- (Levy, 1999) LEVY, P. Inteligência Coletiva. Por uma antropologia do ciberespaço. Lévy, P. Edições Loyola, São Paulo, 1999.
- (Loyal e Bates, 1997) LOYAL, A. Bryan; BATES, Joseph. Personality-Rich Believable Agents that Use Language. Agents'97. Marina del Rey, CA, USA. ACM, 1997.
- (Paiva e Machado, 1998) PAIVA, A.; MACHADO, I. Vincent, an autonomous pedagogical agent for on-the-job training". In: Proceedings of Intelligent Tutor Systems Conference, Springer-Verlag, 1998.
- (Paiva e Machado, 1999) PAIVA, A.; MACHADO, I. Life-long training with Vincent, a web-based pedagogical agent. In: Special Issue on Intelligent Agents for Education and Training Systems of International Journal of Continuous Engineering Education and Life-long Learning, 1999.
- (Paiva et al., 1999) PAIVA, A.; MACHADO, I; MARTINHO, C. The Case of Vincent. In: Proceedings of the AIED Life-like Pedagogical Agents, Le Mans, 1999.
- (Rickel e Johnson, 1998) RICKEL, Jeff; JOHNSON, Lewis. Animated Pedagogical Agents for Team Training. In: ITS'98 WORKSHOP ON PEDAGOGICAL AGENTS, 4. San Antonio: 1998. p.75-77.
- (Shoam, 1997) SHOAM, Yoav An Overview of Agent-Oriented Programming. Bradshaw, Jeffrey (Ed.). *Software Agents*. Menlo Park: AAAI Press/The IT Press, p.271-290.
- (Tows et al., 1998) TOWNS, Stuart; FITZGERALD, Patrick; LESTER, James. Visual Emotive Communication in Lifelike Pedagogical Agents. Proceedings of the Fourth International Conference on Intelligent Tutoring Systems, pp. 474-483, San Antonio, TX, August 1998. (ITS-98)
- (Viccari, 1990) VICCARI, R. Um Tutor Inteligente para a programação em Lógica -Idealização, Projeto e Desenvolvimento, Universidade de Coimbra, 1990. (Tese de Doutorado).
- (Vygotsky, 1998) VYGOTSKY, L. S. A Formação Social da Mente: o Desenvolvimento dos Processos Psicológicos Superiores. Editora Martins Fontes, São Paulo, 1998.
- (Wooldridge e Jennings, 1997) WOOLDRIDGE, Michael; JENNINGS, Nicholas. Intelligent Agents: Theory and Practice. Disponível na Internet. URL: <http://www.doc.mmu.ac.uk/STAFF/mike/ker95/ker95.html.html>.

Layering via Interjeições: possibilidades de detalhamento contextual e progressivo de informações de help

Milene Selbach Silveira^{1,2,3}, Simone Diniz Junqueira Barbosa^{1,3}

¹ Departamento de Informática, PUC-Rio
R. Marquês de São Vicente, 225, Gávea, Rio de Janeiro

² FACIN, PUCRS

³ Tecgraf, PUC-Rio

{milene, sim}@inf.puc-rio.br

Abstract. *This paper proposes an approach for providing better access and content to help systems, designed to clear users' doubts, as expressed by users themselves during interaction. In order to achieve this, we use concepts from Semiotic Engineering, especially communicability, integrated to the notion of layering in help systems.*

Resumo. *Este trabalho propõe uma forma de prover um melhor acesso e conteúdo de help, projetados para esclarecer as dúvidas dos usuários, conforme expresso por eles próprios durante a interação. Para isto, são utilizados conceitos de Engenharia Semiótica, em particular de comunicabilidade, integrados à noção de layering em sistemas de help.*

1. Introdução

A qualidade de um sistema de ajuda (*help*) está relacionada a seu conteúdo e às estruturas de acesso ao mesmo. Devido a isto, os *designers* de *help* deveriam fornecer tanto um acesso facilitado quanto informações claras para que os usuários não apenas “consigam” consultar estes sistemas, mas, também, tenham um retorno produtivo destas consultas.

Em Silveira et al. (2000), foram propostos um modelo e uma arquitetura para sistemas de *help online*, em que o conhecimento do *designer* sobre a aplicação deve ser eliciado durante o processo de desenvolvimento da mesma. Esta proposta está fundamentada nos conceitos da Engenharia Semiótica [de Souza 1993], que visa, entre outras coisas, transmitir, através da interface, a intenção do *designer* e as decisões de projeto: a interface da aplicação é vista como uma mensagem dos *designers* para os usuários. Nesta mensagem, o *help online* é um componente importante, porque é através dele que os *designers* estão mais capacitados a mostrar – diretamente – como eles pensaram e conceberam a aplicação.

Além dos benefícios ganhos com este modelo e arquitetura propostos, sentiu-se a necessidade de investigar mais profundamente as informações que os usuários podem requerer durante (no momento e no contexto de) a interação [Sellen e Nicol 1990]. Um exemplo típico deste tipo de informação contextual pode ser expresso pela pergunta *What's this?*, já disponível em diversas aplicações existentes. Tipicamente, este tipo de acesso leva a uma breve explicação do que é/faz o elemento de interface indicado, sem maiores

aprofundamentos sobre o mesmo. Este acesso costuma ser feito diretamente sobre o elemento em questão.

Já aprofundamentos sobre determinados tópicos de *help* podem ser acessados através de opções do tipo *How?* (Como?), *Related Topics* (Tópicos Relacionados), *Notes* (Observações), *Tips* (Dicas) e âncoras, por exemplo.

Mas estes tipos de informações são suficientemente contextualizadas ao problema do usuário? Como aprofundar esta contextualização? Para melhorar o tratamento das questões contextuais, este trabalho propõe usar as interjeições empregadas no método de avaliação de comunicabilidade [Prates et al. 2000a e 2000b], em diferentes níveis de *affordance* [Norman 1988 e 1999]: operacional, tático e estratégico [de Souza et al. 2000]. Desta forma, consegue-se prover aos usuários uma melhor forma de acessar o conteúdo de *help* quando necessário.

Além desta melhoria na forma e alternativas de acesso, prevê-se um trabalho diferenciado na construção do conteúdo que será apresentado. A idéia geral é apresentar – a cada solicitação do usuário – pequenas porções de informação, numa abordagem minimalista [Carroll 1998], provendo a possibilidade de aprofundamento nos tópicos solicitados, conforme necessidade do usuário.

Nas próximas seções, a abordagem aqui proposta será descrita com base em aplicações de edição de textos¹: os sistemas de *help* das mesmas serão usados como exemplo do que é encontrado atualmente e, também, como base para construção das respostas da nova proposta. Os problemas de interação dos respectivos editores serão mantidos e considerados na construção das respostas de *help*: a idéia, aqui, não é propor melhorias nestas aplicações, e, sim, que o *designer* do *help* reconheça que pode haver problemas e tente prover informações a seu respeito, quando alguma falha de comunicação acontece.

2. Aspectos de Sistemas de Help Tradicionais

Conforme citado no início deste trabalho, informações contextualizadas geralmente resumem-se a descrições de elementos de interface via questões *What's this?* e similares ou acesso a tópicos de *help*.

Quanto a opções do tipo *What's this?*, a abordagem do *StarOffice Writer* é bastante interessante: nela, o usuário pode escolher se quer que as dicas providas pelo sistema, que dão informações operacionais sobre o que é/faz determinado elemento, sejam simples (figura 2.1a) ou estendidas (figura 2.1b), preservando esta opção do usuário durante toda a interação.

¹ Os exemplos aqui apresentados surgiram de uma análise efetuada em sistemas de *help* de aplicações conhecidas de edição de textos: *Microsoft Word*® 97, *Microsoft WordPad*® 98, *Lotus WordPro*® 9.5 e *Sun StarOffice Writer*® 5.2 [Silveira e de Souza 2001].

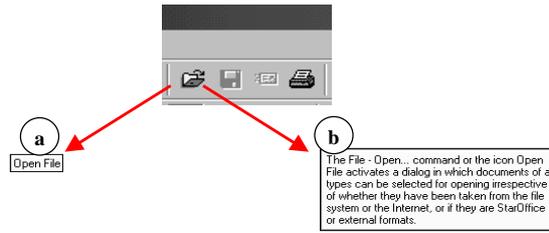


Figura 2.1: Dicas providas para o botão Open (StarOffice Writer) - (a) normal e (b) estendido

Já os tópicos de *help* apresentam, tipicamente, instruções que o usuário deve seguir para utilizar o elemento em questão (figura 2.2).

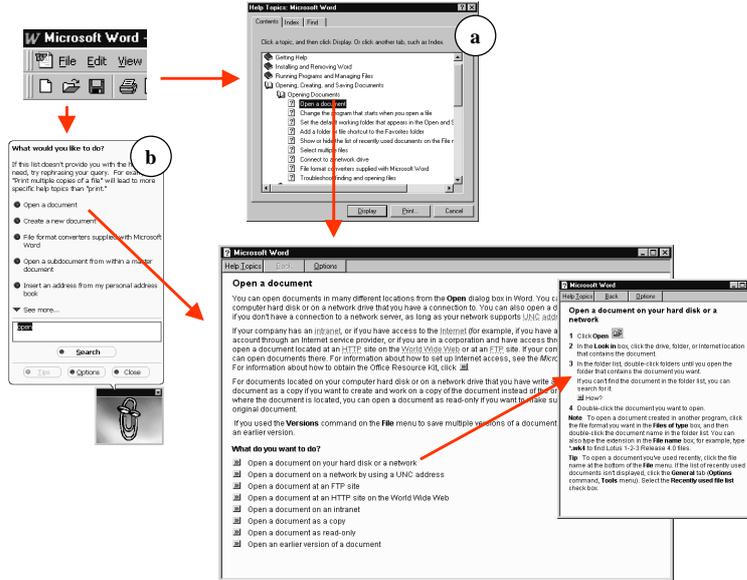


Figura 2.2: Tópico para Open (Word) via (a) tabela de conteúdos ou (b) assistente

Aprofundamentos sobre estes tópicos levam apenas a informações relacionadas (figura 2.3) e demonstrações, sem detalhamentos adicionais.

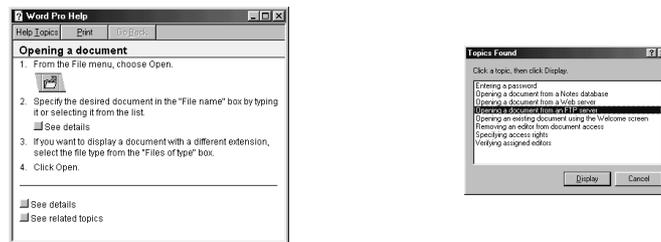


Figura 2.3: Tópicos relacionados para Opening a Document (WordPro)

Estas são as possibilidades de *help* contextual e aprofundamentos encontradas na maioria das aplicações de uso geral. Mas, pode-se notar que nenhuma delas permite um grande aprofundamento sobre o tópico em questão. A maioria trata de assuntos periféricos ou co-relacionados (quanto a mesmo tipo de acesso ou conteúdo fornecido) ou de observações suplementares e dicas.

Mas como prover aos usuários uma maior contextualização, possibilitando que eles mesmo decidam se querem ou não maiores detalhamentos, e de que tipo?

3. Layering como possibilidade de detalhamento contextual de *help*

Segundo Farkas (1998), fazer *layering* significa prover “porções” úteis e bem delimitadas de informação, com canais alternativos de acesso às mesmas, a fim de acomodar diferentes necessidades dos usuários. Este autor ressalta que, em documentações *online*, sua efetividade é maior, dada a possibilidade de criação de *links* de hipertexto. Estes *links* facilitariam a navegação por estes canais e o usuário acessaria informações suplementares somente quando sentisse necessidade das mesmas.

Nas análises efetuadas, foi possível verificar que os *designers* das aplicações proveram alguns tipos de *layering* (intencionalmente ou não). Um dos exemplos mais aparentes de *layering*, encontrados a partir da análise efetuada, são observações suplementares e dicas (*notes* e *tips*²) apresentadas ao usuário dentro do próprio contexto do tópico de *help*. Pode-se presumir que estas sejam informações importantes para uma maior performance e/ou eficácia, sendo, então, dispostas junto à informação principal, para poupar o usuário de mais uma navegação.

O contrário pode ser visto em opções como Tópicos Relacionados, Como?, Ver Detalhes e âncoras. Nestas opções há um botão ou âncora que indica que há mais informações a serem vistas. Por um lado há o incentivo a exploração da interface [Farkas 1998], mas, por outro, presume-se que estas sejam informações adicionais: se o usuário quiser vê-las terá que navegar por elas.

As informações estão compartimentalizadas. Mas será que elas são adequadas à “acomodação das diferentes necessidades dos usuários”, que descrevia Farkas (1998)? Para acomodar as necessidades dos usuários é preciso conseguir auxiliá-lo a realizar a tarefa que

² Aqui refere-se às dicas que são dadas na mesma “tela” que a informação de *help* e não àquelas acessadas via âncoras.

ele se propôs ao começar a interação com a aplicação. É preciso entender o contexto em que ele se encontra.

A resposta a questionamentos do tipo *What's this?* é tratada como uma informação contextual, dado que ela é feita diretamente sobre o elemento de interface desejado. Da mesma forma, então, poder-se-ia dizer que as informações detalhadas (tópicos de *help*) também são contextuais pois, assim como no *What's this?*, o usuário está indicando sobre qual elemento ele deseja informação no momento. Presume-se, desta forma, que, se ele deseja informações sobre o tópico naquele momento, é porque esta é necessária em seu contexto atual de interação.

Mas, o problema destas aplicações é que, por mais que as questões sejam feitas no contexto de trabalho do usuário e as informações sejam dadas de acordo com o que foi por ele solicitado, as respostas obtidas, por mais que tratem do tópico pedido, nunca tratam diretamente da tarefa [Paternò 2000] que ele está tentando realizar ou da dúvida que ele deseja esclarecer.

Ou seja, estas informações solicitadas/obtidas dizem respeito ao contexto da função utilizada pelo usuário e não ao contexto da tarefa que o usuário está tentando realizar. Esta descontinuidade com a situação de uso (contexto da tarefa) na qual o usuário se encontra é o grande problema destas aplicações.

Por fim, quanto aos canais alternativos de acesso às informações, citados por Farkas (1998), o que é possível encontrar nestas aplicações são sistemas de índice ou busca, que levam diretamente a listas de informações disponíveis no sistema, relacionadas ao tópico (função) selecionado. Outra forma alternativa de acesso é através de tópicos relacionados, ressaltando-se neste a dificuldade de se prever o que o sistema considera como sendo um tópico relacionado: informações relativas à mesma função; funções que fazem parte do mesmo escopo; funções que não servem para o mesmo objetivo, mas são acessadas pelo mesmo conjunto de teclas ou botões; etc.

Mas como conseguir prover informações contextualizadas e *layering* efetivo destas informações?

4. Interjeições de Comunicabilidade como Provisão de Acesso e Conteúdo Contextual de *Help*

No método de avaliação de comunicabilidade [Prates 2000a], são usadas interjeições, na tentativa de representar a reação do usuário quando da ocorrência de alguma falha de comunicação durante a interação. Estas falhas ocorrem quando o usuário não consegue perceber as *affordances* do *designer* e, assim, não consegue capturar a mensagem que este está tentando transmitir através da interface da aplicação. Estas *affordances* podem ser de nível operacional, tático e estratégico, segundo de Souza (2000).

A hipótese deste trabalho é que estas interjeições, além de serem usadas para o teste de avaliação de comunicabilidade, podem permitir também uma nova forma de acesso a sistemas de *help*. Os usuários poderiam se expressar através destas interjeições a fim de obter ajuda quando sentissem a ocorrência de alguma falha comunicativa.

Muitas aplicações já provêm acesso a informações específicas de *help* através de expressões *What's this?*, conforme descrito anteriormente. Uma das idéias aqui discutidas é

a possibilidade de estender este conceito a um número maior de interjeições, provendo uma informação contextual de *help* mais relevante a seus contextos atuais de interação.

Por exemplo, para o caso do elemento **Open**, exemplificado anteriormente, seria possível utilizar a interjeição *What's this?* sobre o mesmo e obter uma informação operacional³ (figura 4.1a). Caso fosse desejado um detalhamento maior sobre como executar o mesmo (informação tática), poderia ser usada uma interjeição como *How do I do this?* (figura 4.1b).

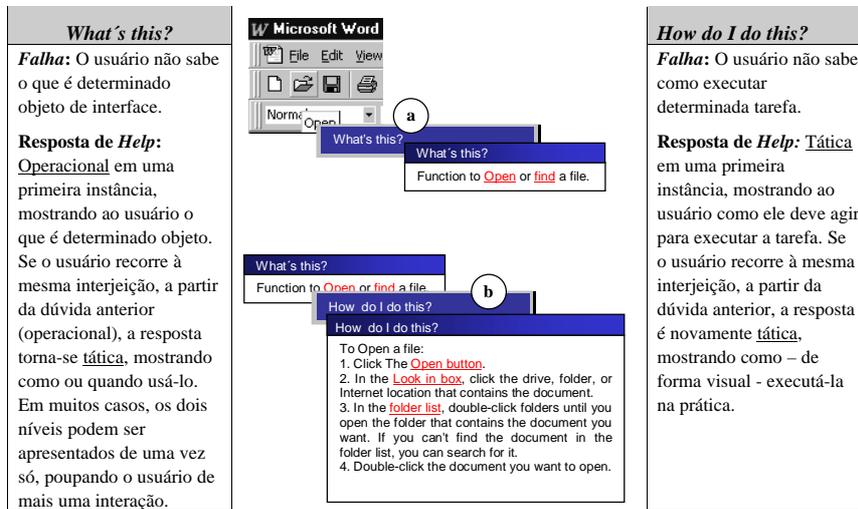


Figura 4.1: Informações via interjeições (a) *What's this?* e (b) *How do I do this?*

Outro exemplo pode ser visto no seguinte cenário, para a função **Track Changes**:

José está fazendo um trabalho em conjunto com Maria e deve concluir o documento em questão, que contém marcas de revisão feitas por ela. Mas ele nunca usou esta função... Ele abre o documento, olha para as marcas de revisão e... O que fazer agora (figura 4.2a)? Ele recebe a resposta de *help*, mas ainda é pouco, ele ainda não sabe como efetuar o que precisa (figura 4.2b). Uhm..., agora entendeu, mas será que não existe uma forma mais simples de fazer isto? (figura 4.2c)

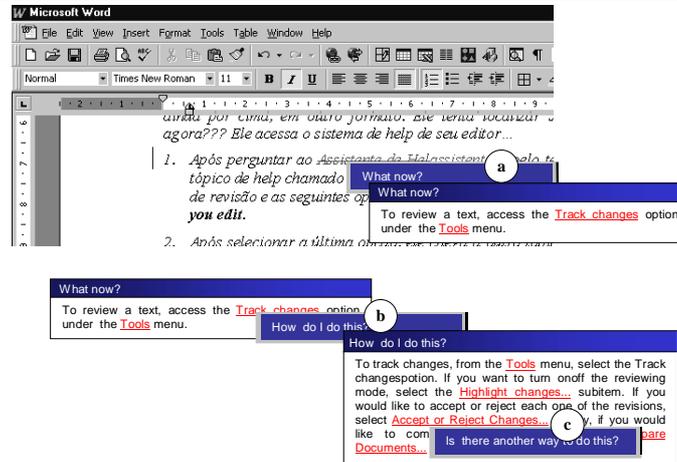
³ Em cada exemplo, são apresentadas as definições das interjeições utilizadas no mesmo. Sempre que, numa resposta de *help* à interjeição, aparecer um item destacado (sublinhado), a partir do mesmo podem ser utilizadas interjeições para acesso ao *help*.

What now?

Falha: O usuário não sabe o que fazer como próximo passo na interação ou o usuário não sabe nem determinar a tarefa que necessita realizar.

Resposta de Help: Operacional em uma primeira instância, mostrando ao usuário o que fazer a seguir (qual seu próximo passo). Se o usuário recorre à mesma interjeição, a partir da dúvida anterior (operacional), a resposta torna-se tática, mostrando como fazê-lo (como dar o próximo passo).

Para o caso de o usuário não saber nem determinar a tarefa que necessita realizar, a resposta torna-se estratégica, apresentando ao usuário as tarefas que a aplicação suporta, de uma perspectiva de usuário, isto é, usando termos com que este esteja familiarizado, de acordo com o domínio correspondente.



Is there another way to do this?

Falha: O usuário sabe como fazer mas quer saber se existem outras possibilidades de caminhos.

Resposta de Help: A resposta do *Help* é tática e estratégica, mostrando os caminhos alternativos possíveis e qual deles é o melhor para cada situação dentro do estado atual da aplicação e sob o ponto de vista do *designer*. Esta interjeição é bastante importante do ponto de vista da Engenharia Semiótica, porque permite ao *designer* expor – explicitamente – suas decisões e intenções de *design*.

Figura 4.2: Informações via interjeições (a) What's now?, (b) How do I do this? e (c) Is there another way to do this?

Outra possibilidade é o usuário não ter nem idéia de como iniciar sua tarefa, conforme o seguinte cenário:

João deseja ordenar uma lista de itens, que ele numerou manualmente, pois nem sabe que existe uma função que faz isto automaticamente. Mas por onde começar? Onde está a função de ordenação (figura 4.3a)? Ele encontra a opção! Vai ao menu, faz o indicado e nada acontece... Estranho! Que será que aconteceu (figura 4.3b)? Ele vai, tenta outra vez e nada... Por que não acontece nada (figura 4.3c)?

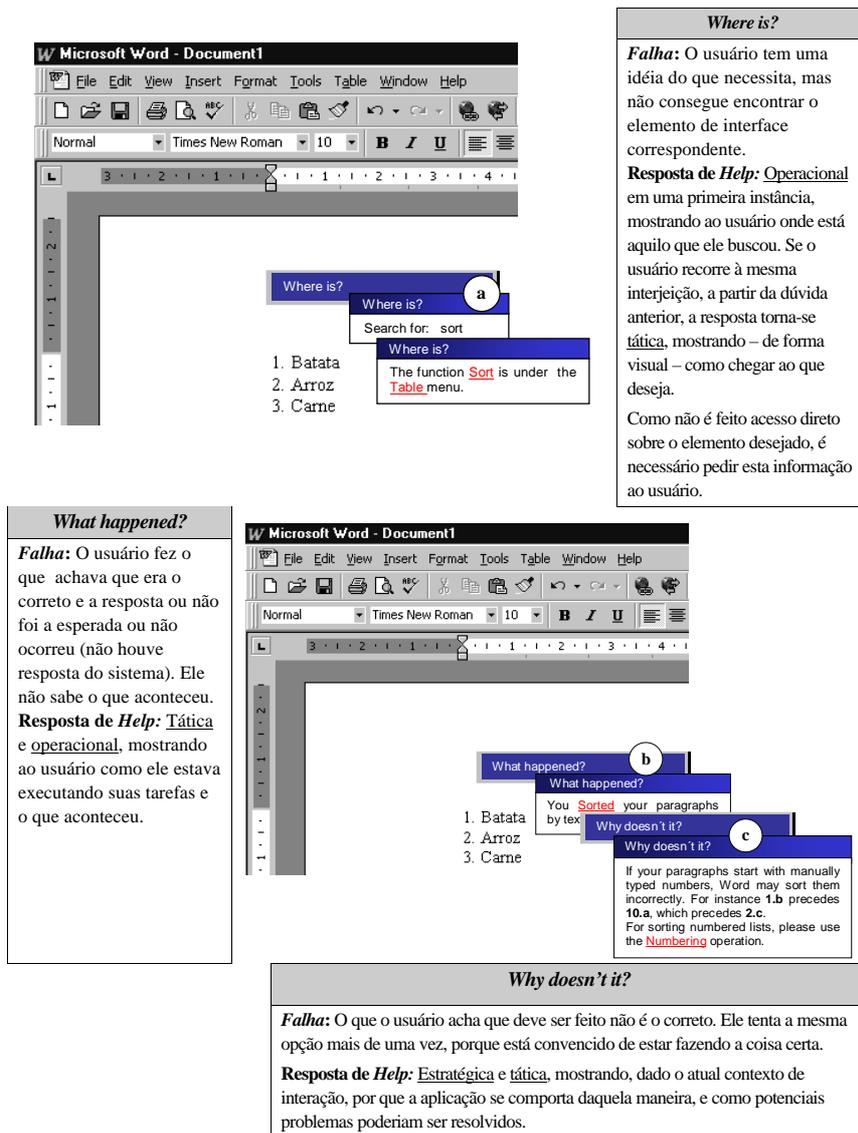


Figura 4.3: Informações via interjeições (a) *Where is?*, (b) *What happened?* e (c) *Why doesn't it?*

Além destas interjeições - vistas nos exemplos anteriores – foram, também, previstas as seguintes (figura 4.4).

<p style="text-align: center;"><i>Oops!</i></p> <p>Falha: O usuário sabe que fez algo errado e quer desfazer a ação que levou ao erro.</p> <p>Resposta de Help: <u>Tática</u>, mostrando ao usuário como ele pode desfazer o que fez, ou seja, a seqüência de ações necessárias para isto.</p>	<p style="text-align: center;"><i>I can't do it.</i></p> <p>Falha: O usuário não sabe como fazer determinada tarefa. Esta interjeição também pode ser utilizada quando o usuário está acessando um conteúdo de <i>help</i> e não sabe como seguir as instruções dadas.</p> <p>Resposta de Help: <u>Operacional</u> e <u>tática</u> em uma primeira instância, mostrando ao usuário o que ele deve fazer e como deve agir. Se o usuário recorre à mesma interjeição, a partir da dúvida anterior, a resposta é novamente <u>tática</u>, mostrando – de forma visual - como executá-la na prática.</p>
<p style="text-align: center;"><i>Help!</i></p> <p>Falha: O usuário quer saber quais as possibilidades de <i>help</i> existentes e como usá-las.</p> <p>Resposta de Help: <u>Operacional</u> e <u>tática</u>, mostrando ao usuário quais as opções de <i>help</i> existentes e como utilizá-las. Se o usuário recorre a mesma interjeição, a partir da dúvida anterior, a resposta é novamente <u>tática</u>, mostrando como executá-la na prática.</p>	<p style="text-align: center;"><i>Where was I?</i></p> <p>Falha: O usuário quer saber “onde ele estava”, ou seja, qual a tarefa na qual ele estava trabalhando anteriormente. Ele quer saber seus passos anteriores para entender o estado em que está no momento.</p> <p>Resposta de Help: <u>Operacional</u> e <u>tática</u>, mostrando a descrição dos passos anteriores do usuário, a partir de seu conjunto de tarefas executado.</p>
<p style="text-align: center;"><i>Why should I do this?</i></p> <p>Falha: O usuário quer saber por que deve fazer determinada tarefa.</p> <p>Resposta de Help: <u>Estratégica</u>, mostrando por que – dentro do contexto da aplicação e sob o ponto de vista do <i>designer</i> - a tarefa deve ser realizada. Do ponto de vista da abordagem semiótica aqui tratada, esta é uma interjeição essencial, porque mostra – diretamente – o ponto de vista do <i>designer</i>. Mostra por que o <i>designer</i> acha que o usuário deve efetuar determinada tarefa e como ela vai influir no contexto geral da aplicação.</p>	<p style="text-align: center;"><i>Who is affected by this? On whom does this depend?</i></p> <p>Falha: O usuário quer saber se executando determinada tarefa a quem ele vai afetar ou de quem ele vai depender.</p> <p>Resposta de Help: <u>Operacional</u>, mostrando quais atores estão inseridos no contexto da tarefa.</p>

Figura 4.4: Demais interjeições

Na tabela 4.1, resumem-se as interjeições aqui propostas. Algumas interjeições de comunicabilidade, apesar de não serem adequadas como forma de acesso ao *help*, ou por serem cobertas por outras interjeições ou por não serem perguntas típicas que o usuário faria, são úteis para a construção de seu conteúdo. Por exemplo, em interjeições como *Looks fine to me.* ou *Thanks, but no, thanks.*, seria o caso de o sistema oferecer ajuda ao usuário quando detecta algum problema por parte dele e não de o usuário pedir ajuda através das mesmas.

E, além das interjeições de comunicabilidade conhecidas, sentiu-se falta de interjeições específicas a esta abordagem (*help*).

Interjeições de Comunicabilidade	Interjeições de Help
<i>Where is?</i> <i>What now?</i> <i>What's this?</i> <i>Oops!</i> <i>I can't do it.</i> <i>What happened?</i> <i>Why doesn't it?</i> <i>Help!</i>	<i>How do I do this?</i> <i>Why should I do this?</i> <i>Who is affected by this?</i> <i>On whom does this depend?</i> <i>Where was I?</i> <i>Is there another way to do this?</i>

Tabela 4.1: Conjunto de interjeições para uso no help

Conforme pode ser visto, algumas interjeições de comunicabilidade podem ser usadas como forma de acesso direto, a partir da interface, como o *What's this?* e *What happened?*. Entretanto, algumas, como o *Where is?*, por exemplo, necessitam que o usuário entre com dados adicionais para que a informação desejada possa ser buscada. Dada a grande quantidade de informações que pode retornar desta busca, o valor da abordagem aqui proposta está em selecionar a informação mais relevante conforme a interjeição usada pelo usuário, de acordo com o visto em Farkas (1998).

Outro item importante é que nem sempre todas as interjeições estarão disponíveis ao usuário. Esta disponibilidade dependerá do elemento em questão. No exemplo a seguir (figura 4.5), baseado no segundo cenário da seção anterior, ao item indicado (elemento sob o *mouse*, no momento do acesso às interjeições) são disponibilizadas apenas parte das interjeições. Por exemplo, não faria sentido um *Oops!* ou um *Why doesn't it?* nesta ocasião, dado que o usuário ainda não fez nenhuma operação no documento (apenas abriu, viu que não sabia o que fazer e acessou o *help*).

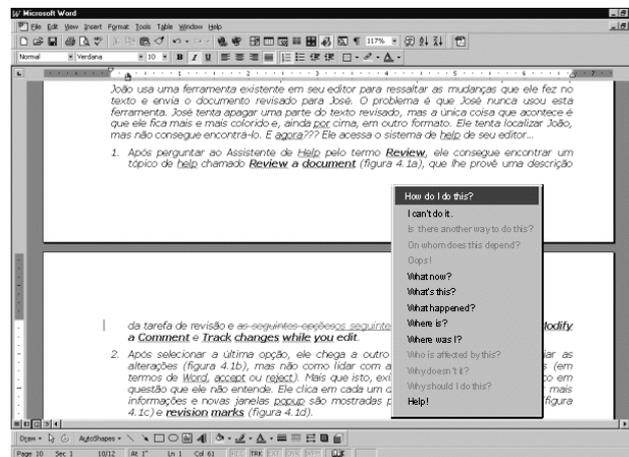


Figura 4.5: Interjeições possíveis, dado o item em questão

Além disso, esta abordagem permite que a aplicação rastreie as dúvidas mais frequentes do usuário e, a partir destas, construa uma lista de FAQ ordenada por um critério real de relevância dado pelos usuários. Este é um caso típico de uso para

interjeições do tipo *Where is?*, *Why doesn't it?* e *What happened?*, por exemplo. Aqui, o usuário não consegue encontrar a informação ou elemento desejado, fica sem saber o que aconteceu na aplicação ou seu porquê. Colocar estas informações na lista de FAQ torna o acesso às mesmas mais rápido no caso de o usuário saber que já teve este tipo de dúvida anteriormente, mas não conseguir se lembrar da resposta correspondente.

5. Considerações Finais

O acesso via interjeições, tanto as do método de comunicabilidade quanto as aqui propostas como interjeições específicas de *help*, permite aos usuários buscar, sob diferentes perspectivas, o que está acontecendo durante a interação. Este novo acesso ao *help*, considerando as previsões – do *designer* – de possíveis falhas que possam ocorrer durante a interação, abre novas possibilidades aos usuários, na tentativa de minimizar suas dúvidas e ajudá-lo com suas necessidades durante a utilização do sistema.

Conforme discutido anteriormente, não adianta apenas um acesso diferenciado. As informações disponibilizadas a partir deste acesso devem ser cuidadosamente construídas a fim de realmente ajudarem o usuário. Conforme visto, existem técnicas de *layering* sendo utilizadas nas aplicações analisadas, mas as mesmas não consideram o contexto da tarefa do usuário. Elas respondem funcionalmente ao problema, muitas vezes deixando o usuário “na mesma” ou “mais perdido” em relação à sua posição anterior. A abordagem de comunicabilidade aqui proposta direciona para o que o usuário quer, quando e como ele quer.

Desta discussão retorna-se a uma questão muito discutida no campo da Interação Humano-Computador (IHC): “Mas, se a interface é boa... Para que é necessário *help*?”. A resposta – a partir da abordagem aqui proposta – é que o usuário precisa entender a aplicação do ponto de vista do *designer* para melhor compreendê-la, não só em termos de suas funcionalidades, mas de seus objetivos. E o *help* é a melhor forma de o *designer* dizer isto ao usuário. Não é apenas dizer como utilizar determinado elemento, mas sim como efetuar determinada tarefa e por que efetua-la dentro do contexto geral da interação.

O trabalho aqui apresentado, bem como o novo modelo e arquitetura de *help online* propostos a partir dele, estão detalhados em Silveira e Barbosa (2001). O desafio atual desta proposta e que está sendo atualmente trabalhado sob orientação de pesquisadores do Departamento de Letras da PUC-Rio é verificar formas eficazes de construir este conteúdo de *help*; testá-lo junto a usuários leigos em IHC e leigos na aplicação; e, testar o modelo e arquitetura em uma aplicação real, em contraste com a abordagem de *help* utilizada atualmente pela mesma.

Referências Bibliográficas

- Carroll, J.M. (ed) (1998) *Minimalism Beyond the Nurnberg Funnel*. MIT Press, Cambridge.
- Farkas, D.K. (1998) Layering as a Safety Net for Minimalist Documentation. In Carroll, J.M. (ed.) *Minimalism Beyond the Nurnberg Funnel*. The MIT Press. Cambridge, MA. pp. 247–274.
- Microsoft. (1995) *The Windows Interface Guidelines for Software Design*. Microsoft Press, Redmond.

- Norman, D. (1998) *The Psychology of Everyday Things*. Doubleday, New York.
- Norman, D. (1999) Affordance, Convention and Design. *Interactions*. Volume 6, Number 3. New York, NY. ACM Press. pp. 38-42.
- Paternò, F. (2000) *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, London, UK.
- Prates, R.O., de Souza, C.S., Barbosa, S.D.J. (2000a) A Method for Evaluating the Communicability of User Interfaces. *ACM Interactions*, 31-38. Jan-Feb 2000.
- Prates, R.O.; Barbosa, S.D.J.; de Souza, C.S. (2000b) A Case Study for Evaluating Interface Design through Communicability. In *Proceedings of ACM Designing Interactive Systems, DIS'2000*. August, 2000.
- Sellen, A. and Nicol, A. (1990) Building User-Entered Online Help. In B. Laurel (ed.), *The Art of Human-Computer Interface Design*. Addison-Wesley, Reading, MA, 1990.
- Silveira, M.S.; Barbosa, S.D.J.; de Souza, C.S. (2000) Modelo e Arquitetura de Help Online. In: *Proceedings of III Workshop on Human Factors in Computer Systems, IHC'2000*. October, 2000. Gramado, RS, Brazil. pp.122-131.
- Silveira, M.S.; Barbosa, S.D.J. (2001) Revelando as *Affordances* do Designer via Sistemas de Help e Interjeições de Comunicabilidade. Série *Monografias em Ciência da Computação* (MCC13/01). DI/PUC-Rio, Rio de Janeiro, 2001.
- Silveira, M.S.; de Souza, C.S. (2001) Análise Comparativa de Sistemas de Help. Série *Monografias em Ciência da Computação* (MCC17/01). DI/PUC-Rio, Rio de Janeiro, 2001.
- de Souza, C.S.; Prates, R.O.; Carey, T. (2000) Missing and Declining Affordances: Are these appropriate concepts? In *Journal of the Brazilian Computer Society*. No.1, Vol.7, July 2000. Sociedade Brasileira de Computação. Rio de Janeiro, RJ, Brazil. pp. 26-34.
- de Souza, C.S. (1993) The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies*, 39, 753-773.

Modelando a Interface de Aplicações em Sistemas de Informação Geográfica como Espaços de Comunicação¹

Juliano Schimiguel, M. Cecília C. Baranauskas, Claudia Bauzer Medeiros

Instituto de Computação Unicamp
Caixa Postal 6176, 13.083-970 - Campinas SP Brasil
{ juliano.schimiguel, cecilia, cmbm }@ic.unicamp.br

Abstract. *The increasing interest by new application interface domains involving Geographical Information Systems (GIS), as for example, urban applications, has motivated the construction of tools that increase the visual reality of these systems. The modeling applications in geographical tools is dependent on the structures and models that they implement making difficult the use of general methodologies for modeling. This paper proposes and discusses the use of a methodology that considers the interface as a Communication Space (Oliveira, 2000), for modeling interfaces of geographical applications. The application of the methodology is illustrated with a specific application domain: the ArcView GIS 3D Analyst.*

Keywords. *Interface as Communication Space, Geographical Information Systems (GIS), Interface Design*

Resumo. *O interesse crescente por interfaces de aplicações geográficas, como por exemplo, aplicações urbanas, tem motivado a construção de ferramentas que permitam um alto nível de realidade visual para tais sistemas. As ferramentas existentes no mercado são fortemente dependentes das estruturas e modelos que implementam e dificultam o uso de metodologias gerais de modelagem. Este artigo propõe e discute o uso da metodologia que entende interface como Espaço de Comunicação (Oliveira, 2000), para modelagem dos aspectos de interface de aplicações geográficas. A aplicação da metodologia é realizada de forma sistemática e orientada ao domínio de uma aplicação específica: o ArcView GIS 3D Analyst.*

Palavras-chave. *Interface como Espaço de Comunicação, Sistemas de Informação Geográfica (SIG), Design de Interface*

1. Introdução

Um dos objetivos principais ao se construir mapas é atribuir significado a informações geográficas (Collinson, 1997). Linhas, pontos e símbolos, entretanto, não se ajustam à nossa maneira de “perceber” o mundo real. Sistemas de Informação Geográfica deveriam considerar um equilíbrio entre as possibilidades da percepção humana por um lado e a realidade espacial por outro, a exemplo do que a própria Cartografia originalmente intencionava (Prado et al., 2000).

Um Sistema de Informação Geográfica (SIG) é um sistema que permite a manipulação, gerenciamento e visualização de dados geo-referenciados. O termo geo-referenciado denota dados que possuem representação em um sistema de coordenadas

¹ Apoio: FAPESP e PRONEX II MCT-SAI

geográficas. Esses sistemas permitem o desenvolvimento de aplicações em que relacionamentos espaciais são importantes. Áreas de aplicação abrangem desde telecomunicações, planejamento urbano ou arquitetura à indústria do turismo e entretenimento (Coors e Flick, 1998).

A maioria dos SIG existentes no mercado são restritos à visão 2D do espaço. A importância de visualização 3D em SIG advém da proximidade entre a interface da aplicação e o mundo real. Representações do mundo real em três dimensões possibilitam aos usuários lidarem com imagens associadas diretamente à sua experiência no mundo. Um dos Sistemas de Informação Geográfica mais difundidos no mercado, o sistema ArcView GIS, possui um módulo denominado ArcView 3D Analyst, para gerenciamento e visualização de dados 3D. O ArcView 3D Analyst permite a criação de interfaces interativas para aplicações que elevam a visualização geográfica e análise para um nível maior de realidade visual.

A habilidade de produzir e visualizar iterativamente idéias de *design* é amplamente reconhecida como tarefa importante em *design* urbano e arquitetura. Embora seja um sistema funcionalmente poderoso, o processo de modelagem de interfaces de aplicações no 3D Analyst pressupõe conhecimento do *designer* para aspectos específicos da tecnologia do sistema. Neste artigo descrevemos a problemática da modelagem de interfaces de aplicações (daqui para frente nomeadas simplesmente modelagem de aplicações) no 3D Analyst e apresentamos como alternativa uma metodologia de modelagem incremental, baseada em Semiótica, que entende a interface como um Espaço de Comunicação (Oliveira, 2000).

O resultado do processo de modelagem de aplicações no 3D Analyst é a *3D scene*, que pode ser caracterizada como um ambiente virtual *desktop*. A *3D scene* permite a visualização e manipulação das entidades representadas em um espaço tridimensional, além disso, possibilita a consulta aos dados associados a essas entidades. O *designer* é quem cria uma *3D scene* e ele não precisa necessariamente ser um especialista em sistemas computacionais e em Sistemas de Informação Geográfica. Entre os tipos de interface existentes, o ambiente virtual é a forma mais avançada de interface com o usuário, sob o ponto de vista de sua possibilidade de imersão. Ele permite sensações de envolvimento, interação e exploração do mundo representado, possibilitando o trabalho com dados extremamente complexos em um ambiente de manipulação mais próximo da prática de percepção do ser humano. Os ambientes virtuais *desktop* surgiram como uma alternativa de baixo custo para os ambientes virtuais altamente aparelhados. Seu desenvolvimento foi impulsionado pelo surgimento da VRML (*Virtual Reality Modeling Language*). Há muitos exemplos de tais ambientes, principalmente voltados para entretenimento. É o caso do Grand Prix II (Microprose, 1996), um ambiente que simula corridas de fórmula 1, tendo grande nível de precisão e reprodução de detalhes de corridas reais.

Neste artigo discutimos o entendimento de interface como Espaços de Comunicação e ilustramos como o processo de modelagem de aplicações 3D, usando SIG, pode ser facilitado dentro dessa perspectiva. O artigo está organizado da seguinte forma: a seção 2 mostra a problemática do processo de modelagem de aplicações no 3D Analyst. A seção 3 define a utilização da metodologia de Espaços de Comunicação como metáfora no *design* de interface de aplicações SIG. A seção 4 ilustra o processo de *design* de interface de uma aplicação geográfica como Espaço de Comunicação. Por fim, a seção 5, apresenta conclusões do trabalho.

2. A Problemática do Design de aplicações no 3D Analyst

Na grande maioria das vezes, os Sistemas de Informação Geográfica são utilizados por especialistas em aplicações que gerenciam informação geográfica (cartógrafos, geógrafos, técnicos em geoprocessamento), não necessariamente habituados à complexidade inerente ao processo de modelagem de tais aplicações.

A construção de aplicações no 3D Analyst é baseada no conceito de projeto, para possibilitar o gerenciamento do desenvolvimento de uma aplicação. Um projeto pode conter vários elementos: *views*, *tables*, *charts*, *3D scenes*, etc. Uma *view* é um mapa interativo representado em duas dimensões, que permite visualizar, explorar, consultar e analisar dados geográficos. Esses dados são organizados em temas, e cada um deles descreve uma parte específica da interface da aplicação. Exemplos de temas são: relevo, hidrografia, construções, estradas, entre outros. As *tables* são estruturas para armazenamento de dados dos temas. Os *charts* são representações visuais de dados das *tables*, que podem facilitar a visualização de informações. A *3D scene* é um ambiente virtual que permite a visualização 3D e consulta de dados geográficos. Sobre ela, podem ser realizados vários tipos de operações, como por exemplo, navegação, rotação, seleção, *zoom*; além disso, a *3D scene* pode ser exportada para o formato VRML, para ser visualizada na *Web* (Manual ArcView 3D Analyst, 1997).

Quando o *designer* inicia a modelagem da interface de aplicação SIG, o primeiro passo é adicionar temas à *3D scene* que estará resultando desse processo. Os temas podem ser de natureza 2D e 3D. Os temas 2D, quando representados no espaço 3D, serão inicialmente visualizados no plano. Posteriormente, tais temas devem ser trabalhados pelo *designer* para que sua representação adquira formas características de um mundo 3D. Algumas das operações envolvidas nesse processo são: criação do efeito de elevação, atribuição de altura-base, criação de modelos de superfície², entre outros. Uma *3D scene* pode, também, ser criada a partir de uma *view*.

Para ilustrar o processo, consideremos inicialmente, o caso de modelagem da interface de uma aplicação a partir de uma *view* e com o seguinte conjunto de temas voltado para planejamento urbano: elevação de terreno, estradas, perímetro de região, um prédio em construção, casas e prédios em construção. Na *view* em questão, temos a representação 2D da interface da aplicação. Essa aplicação deve ser, então, convertida pelo *designer* para uma *3D scene*. A Figura 1 mostra o resultado desse processo.

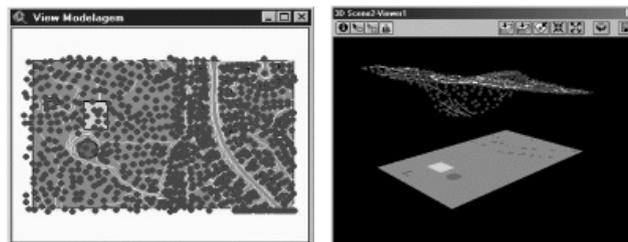


Figura 1. Representações planar (*view*) e 3D da interface da aplicação (*3D scene*)

² Superfícies comuns a aplicações geográficas são do tipo “malha”, em que pontos são espalhados segundo alguma distribuição. Malhas do tipo *grid* (grade retangular) utilizam distribuição uniforme e regular de pontos. Malhas do tipo *TIN* (grade triangular) representam a superfície usando distribuição irregular de pontos.

Na continuidade do processo de modelagem, o *designer* precisa de conhecimentos que extrapolam o seu domínio de atuação. Por exemplo, para atribuir o efeito de elevação a temas, é necessário utilizar o recurso de *extrusão* (Ormsby e Alvi, 1999). Com esse recurso, pode-se gerar linhas verticais a partir de pontos, caminhos verticais a partir de linhas e blocos 3D a partir de polígonos. A *extrusão* não é um processo simples, pois o *designer* precisa relacionar conceitos e estruturas do sistema, como expressões e campos de tabela de tema, ao objetivo que tem em mente.

Uma outra dificuldade surge quando o *designer* precisa colocar a representação dos temas em um mesmo nível de superfície. Como se pode verificar na *3D scene* da Figura 1, o ArcView GIS 3D Analyst representa os temas de natureza 2D espacialmente separados dos temas com informação 3D. Para colocar a representação dos temas em um mesmo nível, o *designer* terá que utilizar um recurso denominado *altura-base*. Novamente, ele precisará entrar em contato com conceitos não familiares, como por exemplo, expressões, campos de tabela de tema e superfícies do tipo *grid* e *TIN*. A Figura 2 ilustra o resultado do processo de modelagem após a aplicação dos recursos de *extrusão* e *altura-base*.

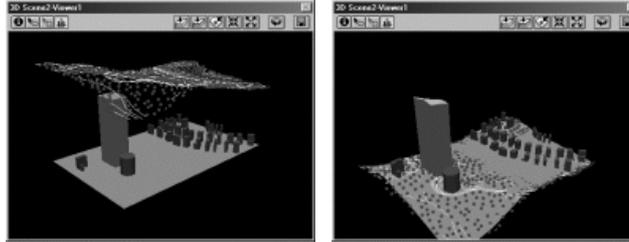


Figura 2. Representação da interface da aplicação após o uso das operações de *extrusão* e *altura-base*

O processo descrito ilustra alguns dos conceitos necessários ao *designer* na modelagem da dimensão 3D de dados. A seção a seguir introduz a metáfora de interface como um Espaço de Comunicação, que permite abstrair parte da complexidade do processo de modelagem de tais aplicações.

3. Espaços de Comunicação como metáfora no design de aplicações SIG 3D

A abordagem adotada neste artigo considera a interface como um mundo habitado por várias entidades, humanas e computacionais, com capacidade comunicativa. O entendimento da comunicação existente na interface apoia-se em princípios da Semiótica.

A Semiótica é a ciência que estuda os signos e a sua vida na sociedade (Peirce, 1990; Santaella, 1996). Um signo é definido como uma relação triádica entre os elementos: o *objeto*, o *representamen* e o *interpretante*. O *representamen* representa, em algum meio, o *objeto* do mundo. Um processo de significação ou semiose ocorre quando, ao se apresentar o *representamen* a uma mente, nela é produzida uma idéia que remete ao *objeto*; esse processo mental é chamado *interpretante* (Prado e Baranauskas, 2000; Prado et al., 2000).

A aplicação da Semiótica no *design* de software permite considerar não somente fatores imediatos relacionados à Interação Humano-Computador, mas também questões sociais e culturais existentes nessa comunicação. Em particular, a abordagem semiótica tem

mostrado resultados relevantes no domínio de interfaces para aplicações em Sistemas de Informação Geográfica (Prado e Baranauskas, 2000; Prado et al., 2000; Prado, 2001).

O usuário de uma aplicação 3D, segundo a perspectiva semiótica, é considerado uma das entidades participantes desse ambiente, sob o ponto de vista de imersão em alguma entidade da interface. O conceito de imersão é utilizado de forma natural em diversas áreas do conhecimento. Tomando como exemplo os cursos para aprendizado de língua estrangeira, alguns dos métodos utilizados defendem que um idioma não pode ser aprendido somente dentro da escola, com um professor, mas que é necessário sua prática em situações cotidianas. Isso é obtido com uma total imersão (real ou virtual) no país onde se fala o idioma em questão. Na modelagem de aplicações SIG, a imersão possibilita obter informações que não seriam captadas se visualizássemos o cenário apenas como espectadores. Para cada entidade em que o usuário imerge, há um cenário diferente do ambiente virtual, permitindo captar signos não observados em outros cenários.

3.1 Uma instanciação da metáfora no 3D Analyst

Segundo a abordagem semiótica em questão, toda entidade no mundo da interface possui uma certa capacidade de emissão de signos e de semiose relativa ao interpretante humano.

Se aplicada ao 3D Analyst, corresponde em avaliar as entidades de uma *3D scene* quanto à emissão de signos. Isto significa verificar se elas comunicam algo a outras entidades, através de propriedades como cor, textura, tom, ruído, cheiro, ação ou por meio de sentenças em uma linguagem conhecida pelo usuário. É este tipo de comunicação que permite ao usuário da *3D scene* reconhecer e interpretar o significado de entidades como ruas, construções, árvores, entre outros.

A capacidade de semiose das entidades, percebida segundo o ponto de vista de um usuário imerso, pode ser ilustrada nos exemplos a seguir:

- a *3D scene*, ela própria como entidade da interface, “percebe” que determinados temas foram selecionados, conseqüentemente, eles devem estar visíveis.
- certas entidades devem “perceber” que, após uma operação de navegação ou *zoom*, feita pela entidade humana, elas devem ser ocultadas, por ficarem fora do seu campo de visão.
- certas entidades devem “perceber” que receberam uma entrada de valor de elevação e devem modificar a sua forma.
- uma entidade *A* “percebe” que há outra entidade *B*, e que deve ser colocada na mesma altura que esta, a partir de uma entrada da entidade humana.
- a entidade humana percebe sua localização na *3D scene*.
- a entidade humana percebe que são necessárias mudanças na estrutura planejada, a fim de se criar outro protótipo no processo de modelagem da aplicação 3D.
- a entidade *3D scene* percebe que a entidade humana navegou pelo ambiente, devendo gerar a imagem referente à nova posição.

Na abordagem proposta, portanto, o *design* da *3D scene* resulta da comunicação entre as entidades da interface. Essa comunicação é estabelecida através de uma “linguagem local”, que tende continuamente a ser estabelecida, pelas relações advindas entre as entidades (Oliveira, 2000). A entidade humana tem sua linguagem vinculada aos tipos de interações possíveis na *3D scene*, como navegação, *zoom*, seleção ou qualquer

outra forma de manipulação. As entidades não humanas “comunicam” através de sua cor, forma, textura, etc.

A seção a seguir ilustra essa nova perspectiva no processo de modelagem de interface de aplicações SIG 3D.

4. Modelando a Aplicação como um Espaço de Comunicação

Na metodologia utilizada neste trabalho, o *design* é entendido como um processo cíclico onde protótipos do ambiente virtual são continuamente desenvolvidos, usados e avaliados. Esta metodologia tem três etapas inter-relacionadas: Análise de Alternativas, Desenvolvimento e Avaliação, conforme ilustrado na Figura 3.

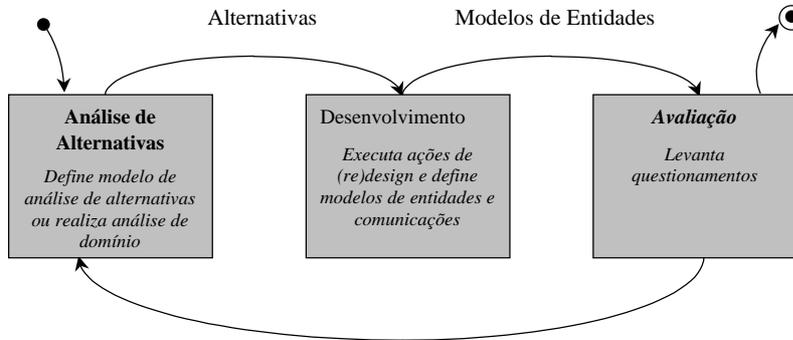


Figura 3. Fases do processo de modelagem

O ponto de partida do processo é uma análise do domínio da aplicação. Na análise de domínio, o *designer* define quais serão as entidades participantes, assim como a comunicação entre elas. Na fase de desenvolvimento, o *designer* modela a interface da aplicação, de acordo com os conceitos definidos na fase de análise de alternativas. Na fase de avaliação, podem ser levantadas várias questões, segundo o ponto de vista do usuário imerso no ambiente virtual, de acordo com a abordagem semiótica em questão. Essas questões são importantes na fase de análise de alternativas do próximo ciclo. Uma questão de *design* é levantada sempre que houver inconsistência entre o *interpretante* presumido para um usuário imerso no ambiente e o interpretante do *designer*, com relação ao *representamen* da entidade. Na fase de análise de alternativas, para cada questão levantada no ciclo anterior do processo, devem ser propostas alternativas de solução e um conjunto de argumentos contra e/ou a favor de cada alternativa; o ciclo se repete, então, em nova iteração.

4.1 Modelos aplicados ao Processo

Nas fases de desenvolvimento e avaliação, há dois modelos que abstraem conceitos da comunicação entre entidades: o modelo de entidades e o modelo de comunicações. Na fase de análise de alternativas, há um modelo de análise de alternativas para tratar e materializar esses conceitos, gerando alternativas para (re)*design*.

O modelo de entidades tem por objetivo captar os signos de cada entidade e os seus respectivos *interpretantes* presumidos, a partir de cenas do ambiente virtual. Esse modelo

contém um título, a identificação da entidade em que o usuário está imerso, um cenário que ilustra o ambiente virtual (representado por uma imagem ou desenho), o *interpretante* presumido para cada entidade do cenário.

O modelo de comunicações é um conjunto de modelos de diálogo (cenários), que contém um título, a identificação da entidade em que o usuário está imerso, uma relação das entidades que participam do cenário. Um cenário pode ser formado por vários trechos, compostos por conjuntos de signos que o descrevem e o *interpretante* presumido para esses signos.

O modelo de análise de alternativas trabalha com questionamentos levantados durante o ciclo de vida anterior da interface de aplicação. Um questionamento é constituído por uma questão e possivelmente por sub-questões derivadas, e um conjunto de alternativas de solução desta questão. Cada alternativa de solução é sustentada por um conjunto de argumentos a favor e é contraposta por argumentos contrários.

Na fase de desenvolvimento, temos como entrada as alternativas selecionadas da fase de análise de alternativas e os modelos de entidades e comunicações da iteração anterior ou da análise de domínio. Esses modelos são re-escritos nessa fase e são usados na fase de avaliação para levantamento de questões de (re)*design*.

Quando realizamos a modelagem, não existe um número pré-definido de modelos a serem criados. Devemos gerar tantos modelos quanto forem necessários, mas cada um deles deve ser um modelo saliente, ou seja, um modelo que capte aspectos não presentes em outros modelos.

4.2 Uma instância da metodologia no 3D Analyst

Para realizar a modelagem de interface de uma aplicação no 3D Analyst, segundo a metodologia apresentada, consideraremos o mesmo conjunto de temas já mencionados (elevação de terreno, estradas, perímetro de região, um prédio em construção, casas e prédios em construção), partindo-se de uma *view*. Este exemplo pressupõe 3 ciclos de interação, dos quais ilustraremos o segundo. A Figura 4 mostra os seguintes resultados do ciclo 1: a *view* de aplicação, os questionamentos levantados para o ciclo 2 e os modelos aos quais os questionamentos estão relacionados.

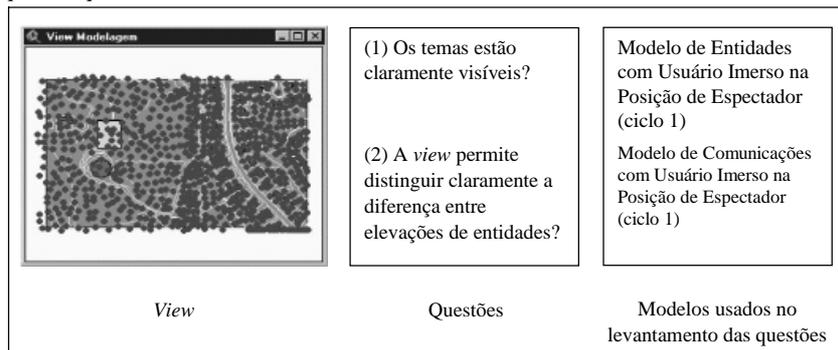


Figura 4. Entrada para Ciclo 2

A) Fase de Análise de Alternativas

A Tabela 1, a seguir, apresenta o modelo de Análise de Alternativas, que subsidia tomadas de decisão para soluções de *design* a serem incorporadas na fase de desenvolvimento.

Modelo de Análise de Alternativas			
Questionamentos	Alternativas	Argumentos	
		Contra	A favor
(1) Os temas estão claramente visíveis?	Criar várias <i>views</i> , de maneira a distribuir os temas entre elas	A divisão dos temas em várias <i>views</i> pode fazer com que o usuário perca muito tempo ao tentar atingir seus objetivos	Por limitações tecnológicas, talvez possa ser interessante para o <i>designer</i> continuar a modelagem em <i>views</i> , pois as <i>3D scenes</i> são mais “caras” computacionalmente, devido ao tempo despendido para renderização, após operações como navegação e zoom
	Criar uma <i>3D scene</i>	Na modelagem 3D, pode-se ter que entrar em contato com conceitos não familiares, como a <i>extrusão</i>	É uma alternativa interessante, pois permite ao usuário visualizar os temas como se estivesse no mundo real
	Restringir o acesso do usuário para seleção de apenas um determinado número de temas por vez.	Não é uma alternativa plausível, pois restringe a liberdade de usuário	Essa é uma alternativa mais “barata” computacionalmente, e evita que se tenha que criar várias <i>views</i> para distribuir os temas entre elas
(2) A <i>view</i> permite distinguir claramente a diferença entre elevações de entidades?	Atribuir cores aos temas, de acordo com o valor de elevação	É uma alternativa viável, mas o 3D Analyst possui recursos para a criação de elevação, como a <i>extrusão</i>	Atribuindo cores aos temas, conseguimos saber quais são as partes mais altas da <i>view</i>
	Criar uma <i>3D scene</i>	A prática com a modelagem em <i>views</i> deve ser modificada para se trabalhar com os conceitos relacionados à modelagem 3D	Parece a solução mais adequada, pois, na modelagem 3D, existem funções específicas para criar o efeito de elevação, como a <i>extrusão</i>

Tabela 1. Modelo de Análise de Alternativas do Ciclo 2

Após a análise dos argumentos contra e a favor, suponhamos que o *designer* tenha selecionado as alternativas “criar a *3D Scene*” para as duas questões levantadas.

B) Fase de Desenvolvimento

As entradas para esta fase são os modelos de entidades e de comunicações da iteração anterior (não ilustrados neste exemplo), e as alternativas selecionadas da fase de análise de alternativas. Essas alternativas são decisões de (re)*design* que, então, são usadas para gerar a *3D scene*, ilustrada na Figura 5:

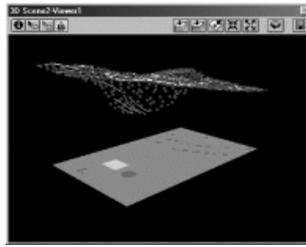


Figura 5. 3D scene gerada a partir da view

Os modelos de entidades e comunicações da iteração anterior são, então, re-escritos. Considerando-se o usuário imerso na entidade de uma construção (Figura 6), obtemos os seguintes modelos de entidades e comunicações (tabelas 2 e 3, respectivamente):

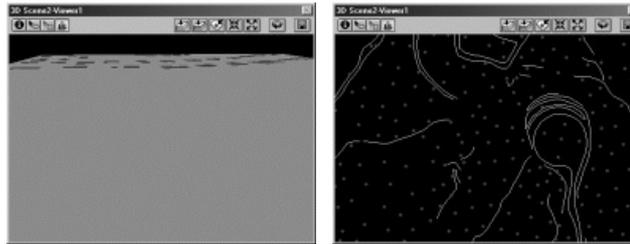


Figura 6. Usuário imerso na entidade de uma construção (olhando para frente e para cima, respectivamente)

Modelo de Entidades				
Título	Entidade em que o usuário está imerso	Cenário	Interpretante presumido para cada entidade	
			Entidade	Interpretante
Modelo de Entidades com Usuário Imerso na Entidade de uma Construção (ciclo 2)	Tema de uma construção	Figura 6	Tema de perímetro	Sugere o perímetro de região (com o usuário imerso olhando para frente)
			Tema de uma construção	É a entidade na qual o usuário está imerso
			Tema de construções	É um conjunto de elementos que sugerem o perímetro de construções (com o usuário imerso olhando para frente)
			Tema de elevação de terreno	É um conjunto de pontos dispersos espacialmente (com o usuário imerso olhando para cima)
			Tema de estradas	É um conjunto de linhas dispersas espacialmente (com o usuário imerso olhando para cima)

Tabela 2. Modelo de Entidades com Usuário Imerso na Entidade de uma Construção (ciclo 2)

Modelo de Comunicações				
Título	Entidade em que o usuário está imerso	Entidades Participantes	Cenário	
			Trechos	Interpretante Presumido
Modelo de Comunicações com Usuário Imerso na Entidade de uma Construção (ciclo 2)	Tema de uma construção	Tema de perímetro e de construções	Em perspectiva, aparecem elementos (retângulos) sobre a superfície	Usuário imerso olhando para frente
		Tema de elevação de terreno e de estradas	Aparece uma seqüência de pontos e linhas	Usuário imerso olhando para cima

Tabela 3. Modelo de Comunicações com Usuário Imerso na Entidade de uma Construção (ciclo 2)

A seguir, a Figura 7, ilustra a *3D scene* resultante da fase de desenvolvimento do ciclo 2. Através dela e dos modelos de entidades e de comunicações definidos, levantamos algumas questões que serão avaliadas na fase de análise de alternativas da próxima iteração (iteração 3).

C) *Fase de Avaliação*

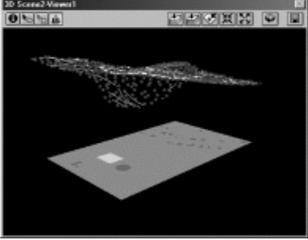
	<p>(1) Os temas de construções são adequados à <i>3D Scene</i>?</p> <p>(2) Os temas estão referenciados a um mesmo nível de superfície?</p>	<p>Modelo de Entidades com Usuário Imerso na Entidade de uma Construção (ciclo 2)</p> <p>Modelo de Comunicações com Usuário Imerso na Entidade de uma</p>
<i>3D Scene</i>	Questões	Modelos usados no levantamento das questões

Figura 7. Entrada para ciclo 3

Concluída essa fase, ocorre uma nova iteração no processo de modelagem. Na fase de análise de alternativas dessa iteração, são levantadas alternativas para os questionamentos, e argumentos contra e a favor a essas alternativas. Após a análise dos argumentos contra e a favor a cada alternativa, foram selecionadas as alternativas: atribuir o efeito de elevação aos temas de construções, com relação ao primeiro questionamento e definir o perímetro de região na mesma altura base do tema de elevação de terreno, com relação ao segundo questionamento. Os modelos de entidades e de comunicações descritos na segunda iteração, juntamente com as alternativas selecionadas na fase de análise de alternativas da terceira iteração, servirão como base para a criação da versão seguinte da interface da aplicação 3D. Depois da iteração 3, foi obtida a versão final da interface da aplicação, que pode ser verificada na Figura 8.

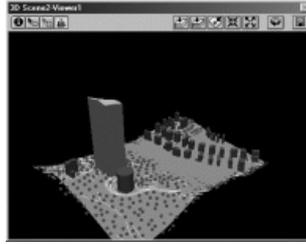


Figura 8. Versão final da interface da aplicação SIG (após iteração 3)

4.3 Análise dos processos semióticos envolvidos na criação de interfaces 3D

Ao contrário do processo *ad hoc* de modelagem em sistemas comerciais, a aplicação da metodologia permite realizar a modelagem de forma sistemática e orientada para os objetos reais do domínio da aplicação. As etapas são organizadas de forma cíclica, com as fases de Análise de Alternativas, Desenvolvimento e Avaliação; cada iteração gera informações importantes para análise no ciclo seguinte: informações de domínio, decisões de (re)*design*, questões de (re)*design*, documentação dos modelos de entidades e comunicações, etc.

Os modelos de entidades e de comunicações são importantes para o levantamento de questões para (re)*design*. As informações para esses modelos são obtidas segundo o ponto de vista de um usuário imerso em várias entidades da interface. Pode-se dizer que, em cada entidade em que o usuário imerge, há um cenário ou uma “realidade” diferente. Em cada um desses cenários, existem signos que não seriam captados em outras perspectivas (com usuário imerso em outras entidades), advindo daí a importância da criação de vários modelos. Em nosso exemplo de modelagem, consideramos alguns modelos que captaram aspectos relevantes no domínio da aplicação, como por exemplo, os modelos de entidades e de comunicações com o usuário imerso na entidade de uma construção (tabelas 2 e 3). Após a fase de análise de alternativas do terceiro ciclo, através do modelo de entidades, com o usuário imerso na entidade de uma construção, o *designer* pôde observar a necessidade de atribuir elevação ao tema de construções e, através do modelo de comunicações, é observada a necessidade de todos os temas estarem referenciados a um mesmo nível de superfície.

A fase de análise de alternativas permite abduzir um conjunto de possibilidades para as questões levantadas em cada fase de avaliação. Além disso, contribui com a avaliação dessas possibilidades, a fim de gerar as alternativas de (re)*design*. A cada iteração desse processo cíclico, teremos versões mais especializadas da interface da aplicação 3D e mais próximas da realidade do mundo que percebemos.

5. Discussão e Conclusão

Ambientes virtuais têm sido criados para aplicações específicas, com suas próprias instâncias e objetivos, técnicas de interação e aparelhagem. Para a modelagem de aplicações no 3D Analyst, precisamos de uma metodologia típica para ambientes virtuais *desktop*, pois eles são uma alternativa eficiente e de baixo custo, em relação aos ambientes virtuais altamente aparelhados. Segundo Darken e Sibert (1996), um dos componentes da realidade virtual é a inserção de humanos dentro desses ambientes. Esses são, exatamente, alguns dos preceitos da metodologia de Espaços de Comunicação.

A modelagem de aplicações geográficas é *ad hoc* e não segue padrões ou métodos estabelecidos; as ações de *designer* são dependentes do conhecimento da tecnologia subjacente a SIG. A metodologia adotada neste trabalho permite uma abordagem sistemática ao processo de criação de interfaces de aplicações 3D, a partir do conhecimento do *designer* sobre a realidade a ser modelada.

Foge ao escopo deste trabalho um estudo comparativo com outras abordagens metodológicas, mas o uso da metodologia Espaços de Comunicação se explica porque uma de suas principais características é possibilitar ao próprio *designer* considerar a experiência da imersão. Com a imersão em alguma entidade do ambiente virtual, o *designer* capta signos não observados em outras imersões, podendo identificar as inconsistências entre o seu *interpretante* para as entidades em relação ao *representamen* dessas entidades. Isso é importante para o levantamento de questionamentos que serão analisados para a tomada de decisões de (re)*design*.

Como próxima etapa do trabalho, estaremos desenvolvendo sobre o sistema ArcView GIS 3D Analyst, uma camada de interface para modelagem de aplicações 3D que implementa a metodologia descrita. Além de documentar o processo e automatizar as ações de (re)*design*, é possível abstrair a complexidade inerente da modelagem de aplicações SIG 3D e aproximar a tarefa da nossa realidade percebida.

6. Referências

- Collinson, A. (1997) "Virtual Worlds. The Cartographic Journal", vol. 34, n° 2, pp. 117-124.
- Coors, V. e Flick, S. (1998) "Integrating Levels of Detail in a Web-based 3D-GIS", ACM GIS'98, pp. 40-45.
- Darken, R.P. e Sibert, J.L. (1996) "Wayfinding strategies and behaviors in large virtual environments", in Human Factors in Computing Systems, CHI'96 Conference Proceedings, pages 142-149.
- Manual ArcView 3D Analyst (1997) "ArcView 3D Analyst. 3D Surface Creation, Visualization, and Analysis", ESRI - Environmental Systems Research Institute, Inc.
- Microprose (1996) "Manual do Usuário do Grand Prix II - World Circuit Racing".
- Oliveira, O.L. (2000) "Design da Interação em Ambientes Virtuais: uma abordagem semiótica", tese de doutorado. Instituto de Computação, UNICAMP. Campinas, SP.
- Ormsby, T. e Alvi, J. (1999) "Extending ArcView GIS. Network Analyst, Spatial Analyst and 3D Analyst", published by ESRI - Environmental Systems Research Institute, Inc.
- Peirce, C.S. (1990) "Semiótica", ed. Contexto. Tradução de Collected Papers of Charles Sanders Peirce.
- Prado, A.B. (2001) "Semiótica e Interfaces de Sistemas de Informação Geográfica", tese de mestrado. Instituto de Computação, UNICAMP. Campinas, SP.
- Prado, A.B. e Baranauskas, M.C.C. (2000) "Avaliando a meta-comunicação designer-usuário de interface", anais do III Workshop sobre Fatores Humanos em Sistemas Computacionais. Gramado, RS.
- Prado, A.B.; Baranauskas, M.C.C. e Medeiros, C.M.B. (2000) "Cartography and Geographic Information Systems as Semiotic Systems", in Proc. 8th ACM GIS International Symposium, Washington D.C., USA.
- Santaella, L. (1996) "O que é Semiótica", 12.ed.. São Paulo: Editora Brasiliense.

O Uso de Técnicas de Visualização de Informação como Subsídio à Formação de Comunidades de Aprendizagem em EaD

Luciana Alvim Santos Romani¹, Heloísa Vieira da Rocha²

¹Embrapa Informática Agropecuária
Caixa postal 6041 - CEP 13083-970 - Campinas - SP - Brasil

²Instituto de Computação - Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6176 - CEP 13083-970 - Campinas - SP - Brasil
luciana@cnptia.embrapa.br, heloisa@ic.unicamp.br

Abstract. *The association of communication tools with another resources that allow interaction between participants in the courses, stimulate the building of an e-learning community. However, the interfaces of communication tools generally are textual and sequential, making difficult the process of building of those communities. In the direction of a solution, this paper presents the use of Information Visualization techniques to represent graphically the data generated by communication tools during distance courses.*

Resumo. *Ferramentas de comunicação associadas a outros recursos que permitem a interação entre participantes em cursos a distância, viabilizam a formação de comunidades virtuais de aprendizagem em EaD. Entretanto, o processo de formação dessas comunidades e o acompanhamento de interações é dificultado pela interface (seqüencial e textual) das ferramentas de comunicação geralmente implementadas nos ambientes de EaD. Como uma solução, este artigo propõe a adoção de técnicas de visualização de informação para representar graficamente dados de comunicação gerados nesses cursos.*

1. Introdução

Os ambientes de educação a distância (EaD) têm sofrido várias alterações nos últimos anos, com o objetivo de propiciar a seus usuários, melhores condições de interação a distância. Isto reflete as novas tendências em educação que requerem ajustes no paradigma corrente que utiliza a instrução centrada no professor. Ao longo do tempo, professores vêm tentando deixar de “dar aulas”, passar ou transmitir conhecimentos para compartilhar, trocar e construir para e com os alunos. Nessa nova abordagem, o professor transforma-se do elemento centro-perguntador para um facilitador, condutor das tarefas postas em ação, estimulador, (co)participante, orientador e observador. Além disso, ocorrem também alterações no seu papel relativo à autoridade, poder e controle. Esse novo paradigma de instrução centrada no aprendiz aumenta a ênfase na *interação*.

Trabalho parcialmente apoiado pelo Programa de Apoio à Núcleos de Excelência – PRONEX (CNPq) - Sistemas Avançados de Informações - IC/Unicamp

Segundo Moore [1989 apud Murphy, et al. 2000], existem basicamente três tipos de interação em cursos na Web: aluno-conteúdo, aluno-instrutor e aluno-aluno. A interação é fundamental no processo de aprendizagem, principalmente porque permite construir significados, socialmente, a partir das perspectivas dos participantes, que pode culminar com o desenvolvimento de comunidades virtuais de aprendizagem. As comunidades na Web são agregações sociais que emergem da rede para formar relacionamentos pessoais no "cyberspace". Preece (2000) define comunidade virtual de acordo com os seguintes elementos imprescindíveis:

- **Pessoas** que interagem socialmente à medida em que se empenham para satisfazer suas próprias necessidades ou desenvolvem papéis especiais, tais como líder ou moderador;
- Um **propósito** comum tais como interesses, necessidade, troca de informação ou serviço que forneça uma razão da comunidade existir;
- **Política** na forma de suposições tácitas, rituais, protocolos, regras e leis que guiam as interações das pessoas;
- **Sistemas de computador** para apoiar e mediar a interação social e viabilizar o sentimento de "estar junto fisicamente".

Essas comunidades podem ser formadas a partir de três categorias de atividades de colaboração via Internet: troca interpessoal, coleção e análise de informação, e resolução de problemas. No contexto de cursos na Web centrados no aprendiz e que requerem colaboração, essas três categorias podem ser combinadas. Entretanto, observando experiências recentes de cursos, percebe-se que ainda existem dificuldades referentes ao acompanhamento das interações por parte do professor e à colaboração entre alunos.

Em relação ao design dos ambientes de EaD, nota-se que muitas ferramentas de comunicação têm sido embutidas nos ambientes sem um redesign para adequá-las às necessidades dos usuários (professores e alunos). A maioria dessas ferramentas apresenta informações de maneira seqüencial e textual, o que muitas vezes não é suficiente para a comunicação entre os interlocutores. A interação face a face dispõe de vários recursos que auxiliam no direcionamento da informação desejada para o interlocutor. Alguns desses recursos incluem palavras faladas, entonação do discurso, gestos com as mãos, postura do corpo, orientação, olhar e expressão facial do locutor [Vilhjálmsson e Cassel, 1998]. Contudo, nota-se que os mecanismos disponíveis nas ferramentas não são suficientes para promover a colaboração, pois é preciso que as pessoas estabeleçam um sentimento de confiança mútua. Consequentemente, não se tem visto uma grande interação entre os alunos que ainda sentem necessidade de se remeter ao professor em qualquer situação de dificuldade no curso, sendo que a troca de experiências entre alunos não tem ocorrido. Com isso, os professores recebem uma quantidade excessiva de mensagens, o que dificulta o acompanhamento mais efetivo dos alunos. Uma das hipóteses é de que as ferramentas de comunicação utilizadas nesses cursos ainda não oferecem as pistas necessárias para que haja confiança a ponto de se formar uma comunidade de aprendizagem.

Há urgência, portanto, em se repensar o design dessas ferramentas considerando essas questões. O uso freqüente dos ambientes e a maior familiaridade dos educadores com essa nova tecnologia permite a percepção clara de aspectos negativos e os faz sugerir inovações para aumentar cada vez mais a qualidade dos cursos. E isso pode ser obtido através do desenvolvimento de ferramentas que permitam a visualização da participação e

da interação dos alunos nos cursos, utilizando para isso os dados armazenados através das ferramentas de comunicação.

Para estudar alternativas para esse problema de pesquisa, foi elaborado um modelo de interface gráfica interativa para busca e visualização de informação sobre a participação e a interação nos cursos, que tenha usabilidade e não requeira dos usuários grandes conhecimentos computacionais. Foram utilizadas técnicas da área de Visualização de Informação [Tufte, 1983, 1990; Card et al., 1999], que melhor poderiam representar graficamente o tipo de dado armazenado pelas ferramentas de comunicação. Foi proposta, então, a ferramenta InterMap (Interaction Map) que apresenta uma interface de consulta e visualização, objetivando contribuir para que o professor tenha uma visão mais abrangente das interações no curso e conseqüentemente possa auxiliar os alunos no seu processo de aprendizagem [Romani, 2000].

Este artigo é dividido da seguinte forma: no próximo item é apresentada a área de Visualização de Informação; em seguida é apresentado o Modelo de Interface proposto e finalmente é apresentada a conclusão e trabalhos futuros.

2. Visualização de Informação

A visualização, genericamente, é o uso de imagens para representação de informação significativa [Stasko et al., 1997]. Ela possui vários enfoques diferentes tais como Visualização Científica, Visualização de Informação, Visualização Geográfica, Visualização de Negócios, Visualização Estatística, Visualização de Processo e Visualização de Software. Todos os tipos de visualização compartilham uma meta comum: transformar o dado em algo com mais significado, ou seja, uma representação visual útil de forma que o observador humano possa ter um melhor entendimento. Atualmente, essa transformação é realizada com o auxílio do computador através de recursos gráficos.

Segundo Card et al. (1999), a Visualização de Informação é o uso de representação visual, interativa e suportada por computador, de dados abstratos para ampliar a cognição. O objetivo de representar o dado abstrato visualmente consiste em auxiliar os indivíduos a enxergarem um fenômeno no dado, usando a percepção para diminuir o esforço cognitivo. Isso se dá através da "cristalização do conhecimento" [Card et al., 1999]. Esse processo ocorre quando um indivíduo coleta dados para um propósito específico, analisa-os através de estruturas de representação e então empacota todas essas formas de representação para comunicar a alguém ou tomar uma decisão. Por exemplo, um empregado de uma empresa recebe a tarefa de comprar um determinado equipamento para o seu projeto. Ele faz uma pesquisa de mercado para identificar as características do equipamento de diferentes marcas e cotar os preços, obtendo uma série de dados. Antes de apresentar os dados ainda brutos para o grupo em uma reunião, ele pode montar uma tabela com os produtos organizados nas linhas da tabela e suas características nas colunas. Com essa representação, ele pode identificar padrões entre vários equipamentos, fazer comparações e organizar os dados para apresentá-los mais facilmente durante a reunião, auxiliando na visualização dos dados e na tomada de decisão.

Mas para que a representação visual do dado seja efetiva para as pessoas, é importante que o mapeamento preserve o dado. Card et al (1999) sugerem um modelo de referência para mapear o dado em formas visuais que aplica uma série de transformações no dado bruto tornando-o uma visualização.

2.1. Modelo de referência para visualização

No modelo de referência mostrado na figura 1, a primeira transformação trabalha o Dado Bruto armazenando-o em Tabelas de Dados. Um segundo passo prevê a escolha de uma melhor Estrutura Visual (linear, mapa, tri-dimensional, temporal, árvore) para o dado e finalmente Transformações Visuais como a inclusão de parâmetros gráficos (posição, escala) são adicionados para criar uma visão dos dados.

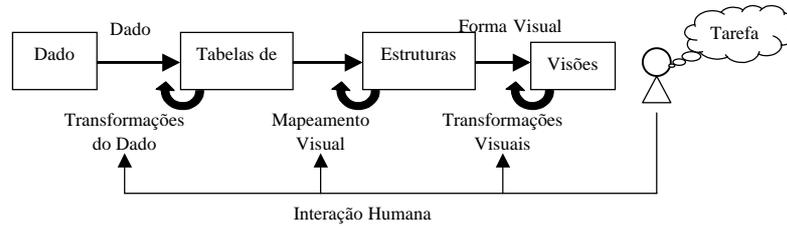


Figura 1: Modelo de referência para visualização (adaptado de Card et al., 1999, p. 17)

2.2. Dado Bruto e Tabelas de Dados

O Dado Bruto é o dado em um formato idiossincrático. Esse dado pode vir em várias formas, desde formulários de computador até textos de livros. Através de transformações de dados que contêm valores derivados ou estruturas, o Dado Bruto é transformado numa relação ou conjunto de relações que são mais estruturadas e mais facilmente mapeadas para formas visuais. Essas relações são denominadas *tuplas*.

O conceito de Tabelas de Dados é diferente daquele atribuído às tabelas em um banco de dados. As Tabelas de Dados combinam relações com metadados que descrevem essas relações. Por exemplo, uma Tabela de Dados para filmes poderia conter os filmes (casos) nas colunas e as variáveis representadas nas linhas conteriam as propriedades daquele filme.

2.4 Estruturas Visuais

Na visualização, Tabelas de Dados são mapeadas para Estruturas Visuais que aumentam o substrato espacial com marcas e propriedades gráficas para codificar a informação. Esse mapeamento pode ser feito de diversas formas. No entanto, bons mapeamentos são difíceis, já que pode ocorrer de dados não desejáveis aparecerem nas Estruturas Visuais. Tufte (1983) afirma que a excelência nos gráficos estatísticos consiste de idéias complexas comunicadas com clareza, precisão e eficiência.

Os gráficos evidenciam o dado e podem ser mais precisos e reveladores do que as computações estatísticas convencionais. X e Y são utilizadas como forma de estabelecer uma referência num plano e pares (x,y) como forma de localizar estruturas visuais num plano destinado a conter os elementos geométricos usados para construir representações gráficas. Na superfície plana (plano bi-dimensional), toda oportunidade para cobrir informação adicional em uma dimensão deve ser considerada cuidadosamente.

Dentro de Estruturas Visuais, as Marcas são os objetos visíveis que ocorrem no espaço. Existem 4 elementos básicos:

- Pontos (0D),
- Linhas (1D ou linear),
- Áreas (2D ou bi-dimensional) e
- Volumes (3D ou tri-dimensional)

Elementos do tipo ponto e linha podem ser associados em uma outra estrutura topológica denominada Grafo e Árvore. Um exemplo do uso dessas estruturas pode ser visto na figura 2 que mostra uma tela do sistema Conversation Map que representa graficamente um grupo de discussão da Usenet¹ [Sack, 2000]. Esse sistema foi projetado para tornar mais fácil para os participantes entenderem e refletirem sobre a conversação de larga escala como a que ocorre em listas de discussão, *newsgroup* da Usenet, etc. O Conversation Map analisa o conteúdo e os relacionamentos entre mensagens e usa os resultados da análise para criar uma interface gráfica. O sistema gera dois grafos denominados, respectivamente, redes sociais (*Social Networks*) e redes semânticas (*Semantic Network*). Nas redes sociais, os nós representam pessoas e as arestas, as trocas de mensagens entre os participantes do grupo de discussão. Nas redes semânticas, os nós representam os termos usados durante as discussões e existe uma aresta entre dois termos se eles foram mencionados na mesma mensagem.

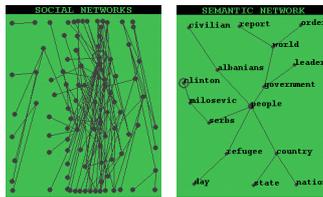


Figura 2: Ferramenta Conversation Map [Sack, 2000 p.2 e p.5]

Esse tipo de representação, através de grafo, também usa a posição para criar noções de proximidade. Como essas características são facilmente captadas pela percepção, é possível codificar informação adicional como por exemplo, a noção de grupo ou direção parcial. Essa técnica é utilizada pela ferramenta VisualWho desenvolvida para a visualização de uma comunidade eletrônica [Donath, 1995]. Utilizando dados de uma lista de *e-mail* é criada uma representação baseada no modelo de padrões de afiliação dentro da comunidade.

A cor é fundamental em Visualização de Informação e o olho humano é extremamente sensível às variações de cor. Embora seja possível que a percepção de cor de um observador possa ser muito diferente da de outro, evidências experimentais sugerem que os relacionamentos entre cores são, em muitos aspectos, relativamente livres de influências culturais e individuais [Jacobson e Bender, 1996]. Historicamente, a cor tem sido caracterizada pela consideração da sua aplicação, que inclui:

- características físicas da cor;
- mecanismos do sistema visual humano;

¹ <http://www.usenet.com>

- aplicações para codificação e reprodução e
- aplicação no design e interatividade.

A aparência da cor é o resultado de uma interação de cores e não é um atributo de representações dentro da mente de um indivíduo. A aparência é definida, discutida e aplicada usando-se a linguagem da percepção. Ver a cor envolve mais do que a sensação da aparência de cores isoladas. É o resultado de uma pluralidade de cores como um todo, independente de como a aparência das cores individualmente é estabelecida. É apenas dentro de um contexto de um conjunto de cores que a cor transmite informação semântica e simbólica. No entanto, deve-se ter cautela no uso das cores, pois é preciso colocar a cor certa no lugar certo, para alcançar um efeito desejado.

2.5. Interação Humana

Para finalizar o modelo de referência, é preciso contemplar a interação humana completando o ciclo entre as formas visuais e o controle dos parâmetros de visualização na realização de alguma tarefa. A forma de interação mais óbvia é a manipulação direta que permite, por exemplo, arrastar um nó de um grafo para o centro da tela [Donath, 1995]. A interação humana com as estruturas visuais e os parâmetros do mapeamento criam um espaço de trabalho de informação que leva ao desenvolvimento de um sentido visual. As técnicas desenvolvidas para a Visualização de Informação envolvem formas de representação do dado (2D, 3D, árvores), interação humana (consultas dinâmicas, manipulação direta) e as tarefas.

3. Ferramenta para visualização da interação (InterMap)

Partindo das necessidades dos usuários, foi proposto um modelo inicial para mapear a interação e a participação dos atores (professor e aluno) envolvidos em um curso a distância. O modelo prevê uma estrutura de pesquisa (simples e de fácil manipulação) através de formulários de consulta e apresentação das informações. A apresentação utiliza técnicas de Visualização de Informação e possibilita ao usuário visualizar:

- dados quantitativos através de *gráficos de barra* (charts);
- a interação dos participantes através de *grafos*;
- a participação em fóruns de discussão através de *representações bi-dimensionais com código de cores*;
- a participação em bate-papos através de uma *representação que evidencia as trocas de turno e pausas*; e
- a obtenção de detalhes através da técnica de *exploração*.

Para validar esse modelo inicial foi desenvolvida a ferramenta InterMap que mapea os dados das ferramentas de comunicação do ambiente TelEduc² - Correio, Fórum de Discussão e Bate-Papo [Romani et al., 2000].

² TelEduc é um ambiente para suporte ao ensino/aprendizagem a distância que vem sendo desenvolvido, desde 1997, pelo Núcleo de Informática Aplicada à Educação (NIED) e Instituto de Computação, ambos da Unicamp.

3.1. Gráficos de Barra

Gráficos de barra são utilizados para mapear informações sobre a participação no curso. Foram implementados gráficos por período ou por participante, definindo-se o tipo de apresentação (gráfico ou tabela); o mês; o ano; o período (por dia, por semana ou por mês) e a categoria (todos os participantes, somente alunos ou somente formadores). Para diferenciar dados de alunos e professores mais facilmente foram utilizadas cores: o cyan para representar os professores e a cor laranja para evidenciar dados de aluno. Com isso, gráficos cuja categoria é *Somente Formadores*, são apresentados em cyan e aqueles que tratam da categoria *Somente Alunos* estão em laranja, como pode ser visto nas duas telas da figura 3.

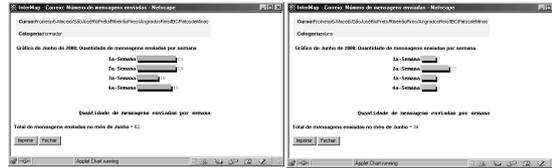


Figura 3: Gráficos por semana para as categorias: *Somente Formadores* e *Somente Alunos*

3.2. Grafos

Para visualizar informações de como se dá a interação em um curso, foi feita a representação dos dados através de um grafo, que é definido de forma simplificada como um conjunto finito e não vazio de vértices e arestas. O usuário define um intervalo de tempo que queira mapear e um grafo é gerado e apresentado em uma outra janela.

Na figura 4 é apresentado um grafo que representa a interação entre os participantes de um curso exemplo. Nesse mapa da interação, os vértices (nós) representam pessoas - isto é, participantes no curso - e as arestas representam a troca de mensagens entre eles. Logo, se o participante "A" envia uma mensagem para o participante "B", uma aresta é desenhada ligando os dois nós chamados "A" e "B". O mesmo ocorre se o participante "B" envia mensagens para o participante "A" ou se ambos enviam mensagens um para o outro. Nessa representação, os nós que não possuem arestas, indicam que tais participantes não enviaram e também não receberam mensagens. O nó "Todos" evidencia quais participantes enviam mensagens para todas as pessoas no curso. Uma outra estratégia para enriquecer o mapeamento seria a utilização de grafo orientado ou dirigido, no qual o conjunto de arestas é formado por pares ordenados de vértices. No entanto, nesse protótipo inicial da ferramenta, como o enfoque principal estava em descobrir a interação entre os participantes através de uma visão geral, optou-se pela não adoção de grafos dirigidos.

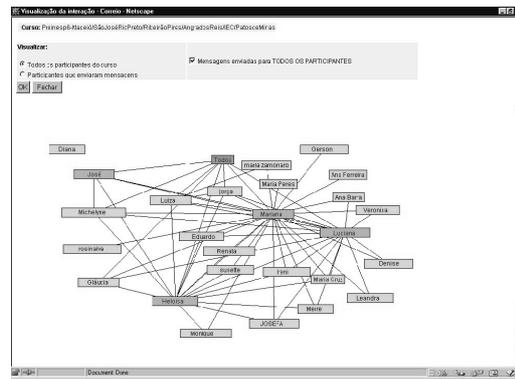


Figura 4: Visualização da Interação

Ao seleccionar um nó do grafo com um clique do *mouse* mantendo-o pressionado, o nó e todas as suas arestas ficam em cor vermelha, facilitando a visualização. Além disso, é possível modificar a apresentação do grafo seleccionando um nó e arrastando-o com o *mouse*. Com isso, o usuário tem liberdade para manipular o grafo distribuindo os nós da forma que melhor lhe convier. Na parte superior da janela, existem alguns controles que permitem alterar a visualização. O uso de controle, permite ao usuário direccionar ou diminuir a informação mapeada.

3.3. Representações bi-dimensionais com códigos de cores

Quando o número de variáveis que se quer mapear é superior a dois, fica difícil representar essa informação através de gráficos de barra. Um exemplo dessa situação é o que ocorre quando se quer saber a quantidade de mensagens enviadas, diariamente, por todos os participantes, individualmente, em um determinado mês. Nesse caso, têm-se 3 variáveis: quantidade de mensagens, participante e dias do mês. Para apresentar visualmente essa informação, pode-se utilizar dois eixos (X e Y) representando o participante e os dias do mês, respectivamente, e a cor para codificar informação sobre a quantidade de mensagens. Com isso, foi proposto um código de cores no qual é utilizada a cor cinza para indicar 1 mensagem, verde para 2 a 3 mensagens, amarelo para 4 a 7 mensagens e vermelho para mais de 7 mensagens, como pode ser visto na figura 5.

De forma diferente dos grafos, o que se pretende com essa representação, é evidenciar a participação dos indivíduos no curso. Com isso, pode-se perceber mais facilmente aquelas pessoas que mais interagem e aquelas que raramente ou nunca participam.

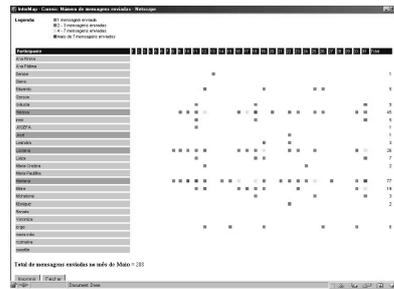


Figura 5: Representação bi-dimensional com código de cores

O mesmo tipo de representação bi-dimensional foi utilizado para mostrar a participação e os padrões sociais na ferramenta Fórum de Discussão. Foram criadas duas formas básicas para evidenciar: De que forma estão estruturadas as respostas? e Como ocorre a intervenção do professor? Na visualização da Estrutura de Respostas, tem-se uma visão geral de um determinado Fórum de Discussão com os participantes representados no eixo X e os assuntos discutidos indicados no eixo Y. Nessa visualização, as primeiras mensagens por assunto são identificadas em verde e as respostas a essas mensagens aparecem em vermelho. Com isso, pode-se avaliar a iniciativa das pessoas em propor assuntos para discussão e a participação no grupo através das respostas. Na visualização da forma de intervenção do professor, as mensagens enviadas ao mesmo Fórum são mapeadas através de pontos em cyan – mensagens enviadas por professores e laranja – mensagens enviadas por alunos. Um exemplo dessas duas visualizações pode ser visto na figura 6.

Para obter detalhes sobre as mensagens, basta selecionar com o *mouse*, a Marca (traços) na Estrutura Visual. Neste caso, uma janela é aberta com o conteúdo da mensagem.

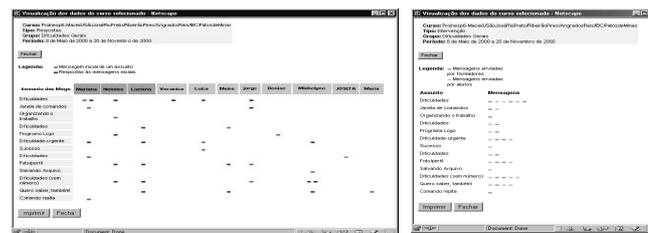


Figura 6.: Representação bi-dimensional (Estrutura de Respostas e Intervenção do Professor)

3.4. Representação que evidencia trocas de turno e pausas

No Bate-Papo, o registro das conversas aparece como uma seqüência de texto, o que ocasiona a perda de componentes não textuais do discurso como pausas e troca de turno que auxiliam na compreensão da discussão [Oeiras e Rocha, 2000]. Para permitir a visualização do fluxo da conversação durante uma sessão de bate-papo é utilizada uma representação visual como a que pode ser vista na figura 7. Nessa visualização, cada

participante é representado pelo seu "apelido" que foi informado no início da sessão. O "apelido" aparece no topo de uma linha na vertical que é de cor laranja quando o participante é um aluno e de cor cyan quando se trata de um professor. A linha aumenta na direção vertical de acordo com o tempo transcorrido da sessão de bate-papo. Cada mensagem enviada é identificada através de um traço na horizontal. O primeiro e o último traços em uma linha representam a entrada e a saída de um participante da sessão de bate-papo, respectivamente. As mensagens podem ser visualizadas em uma outra janela quando o usuário seleciona um traço com o *mouse*.

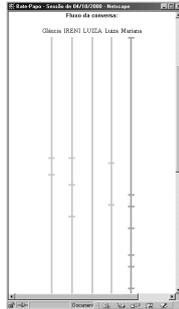


Figura 7: Visualização de uma sessão de bate-papo (trocas de turno e pausas)

3.5. Exploração

A técnica de Exploração, mostra detalhes revelando informação adicional sobre uma determinada Marca (ponto, reta, grafo, etc.). Essa técnica foi utilizada para que o usuário pudesse obter detalhes sobre os participantes do curso ou sobre as mensagens enviadas (figura 8). Para tanto, basta selecionar com o *mouse*, o nome ou ponto desejado na Estrutura Visual. Com isso, são apresentadas, em uma outra janela, informações detalhadas sobre a pessoa ou sobre a mensagem selecionada.

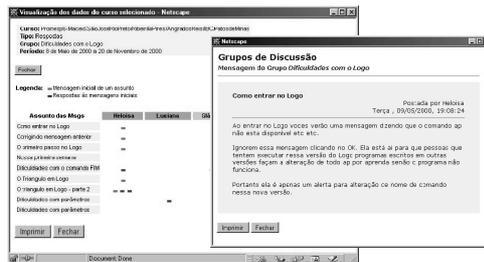


Figura 8: Janela com detalhes do conteúdo de uma mensagem

4. Conclusões

Este trabalho apresenta um modelo inicial de interface para visualização de informações sobre as interações e participações que ocorrem em ambientes de educação a distância na Web. O modelo tem a vantagem de permitir aos usuários um acesso rápido às informações sem o ônus de se ter que conhecer a sintaxe de uma linguagem complexa para usuários

leigos como a SQL (Structured Query Language) e a estrutura do banco de dados usado pelo ambiente. Além disso, as visualizações apresentadas evidenciam a participação e a interação dando aos usuários uma visão geral da massa de dados ou detalhes sob demanda, o que facilita sua compreensão do que está ocorrendo no curso em tempo hábil para uma tomada de decisão.

As técnicas de Visualização de Informação utilizadas neste trabalho oferecem uma alternativa viável para busca, recuperação e representação de dados em ambientes de educação a distância. Essas técnicas puderam ser implementadas de forma simplificada, sem provocar sobrecarga no ambiente e ainda assim produzir bons resultados.

Através dos testes de validação realizados pôde-se constatar que as visualizações disponíveis na ferramenta auxiliam o professor no acompanhamento dos alunos. Entretanto, novas formas de visualização precisam ser estudadas para facilitar o uso das ferramentas de comunicação beneficiando os participantes durante a interação.

Para contemplar e possibilitar uma análise mais qualitativa dos dados, poderiam ser aplicados conceitos e aspectos da área de Inteligência Artificial para o desenvolvimento de agentes³ inteligentes. Esses agentes poderiam analisar o mapeamento dos dados e identificar os indivíduos que participam muito, mas não contribuem de forma efetiva para o curso. Para isso, esses agentes poderiam utilizar um conjunto de termos relevantes ao conteúdo do curso organizados em uma base, por exemplo. Os agentes também poderiam auxiliar na busca de informação relevante, identificando especialistas em assuntos do curso.

Além disso, pode-se usar técnicas de Visualização de Informação para uma análise do discurso. Na literatura encontram-se alguns trabalhos que apresentam uma análise semântica sobre os dados de grupos de discussão [Sack, 2000]. No entanto, ainda são poucos os resultados principalmente em ambientes educacionais na Web.

De forma geral, conclui-se que o uso de técnicas de Visualização de Informação para mapear dados de interação e participação nos cursos a distância na Web, configura-se um passo importante para auxiliar o professor a realizar um melhor acompanhamento dos alunos e conseqüentemente contribuir para a melhoria do processo ensino/aprendizagem. Além disso, aplicando essas novas formas de representação pode-se diminuir o esforço cognitivo das pessoas envolvidas em um curso e dar subsídios para auxiliá-las a tomar conhecimento do mundo social no curso.

Referências

- Card, S.K., Mackinlay, J.D., Shneiderman, B. (1999) *Readings in information visualization: using vision to think*. San Francisco, California: Morgan Kaufmann Publishers. 686p.
- Donath, J. (1995) Visual Who: animating the affinities and activities of an electronic community. In: ACM MULTIMEDIA 95, San Francisco, California. *Electronic Proceedings*.

³ Agente é definido como uma entidade computacional que age no lugar de outra entidade de uma forma autônoma; realiza suas ações com um certo nível de pró - atividade e/ou reatividade; e possui um certo nível de aprendizagem, cooperação e mobilidade [Green et al., 1997].

- Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., Evans, R. (1997) *Software agents: a review*. Disponível: http://www.cs.tcd.ie/research_groups/aig/iag/toplevel2.html Consultado em 02 nov. 2000.
- Jacobson, N., Bender, W. (1996) Color as a determined communication. *IBM Systems journal*, MIT Media Lab, v.35, n.3/4, May. Disponível: <http://www.research.ibm.com/journal/sj/mit/sectiond/jacobson.html> Consultado em 06 nov. 2000.
- Moore, M. G. (1989) Three types of interaction. *The American Journal of Distance Education*, v.3, n.2, pg. 1-6.
- Murphy, K. L.; Mahoney, S. E.; Harvell, T. J. (2000) Role of contracts in enhancing community building in Web courses. *Educational Technology & Society*, v.3, n.3.
- Oeiras, J.Y.Y., Rocha, H.V. da (2000) *Uma modalidade de comunicação mediada por computador e suas várias interFACES*. In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 3, Gramado. *Anais*. p.151-160.
- Preece, J. (2000) *Online Communities - Designing Usability, supporting sociability*. Chichester: John Wiley & Sons. 439 p.
- Romani, L. A. S. (2000) *InterMap: ferramenta para visualização da interação em ambientes de educação a distância na Web*. Campinas: Instituto de Computação da UNICAMP. 116p. (Dissertação, Mestrado em Ciência da Computação).
- Romani, L. A. S.; Rocha, H. V. da; Silva, C. G. da (2000) Ambientes para educação a distância baseados na Web: Onde estão as pessoas? In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 3. *Proceedings*. Gramado, Brasil, outubro.
- Sack, W. (2000) Conversation Map: a content-based usenet newsgroup browser. In: INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES, New Orleans, LA. *Proceedings of the 2000 international conference on Intelligent user interfaces*. 233-240p.
- Stasko, J.; Domingue, J.; Brown, M.H.; Price, B. A. (1997) *Software visualization: programming as a multimedia experience*. Massachusetts: MIT Press. 562 p.
- Tufte, E. R. (1983) *The visual display of quantitative information*. Cheshire, CT: Graphics Press. 197p.
- Tufte, E. R. (1990) *Envisioning information*. Cheshire, CT: Graphics Press. 126p.
- Vilhjálmsón, H. H., Cassel, J. (1998) BodyChat: autonomous communicative behaviors in avatars. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 2. *ACM Proceedings*. p.269-276

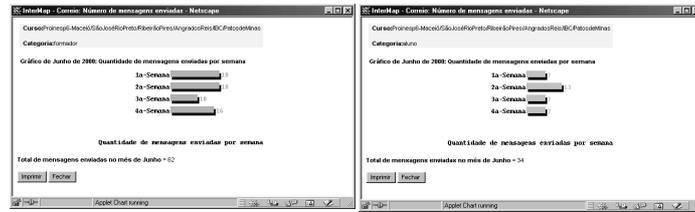


Figura 3: Gráficos por semana para as categorias: Somente Formadores e Somente Alunos

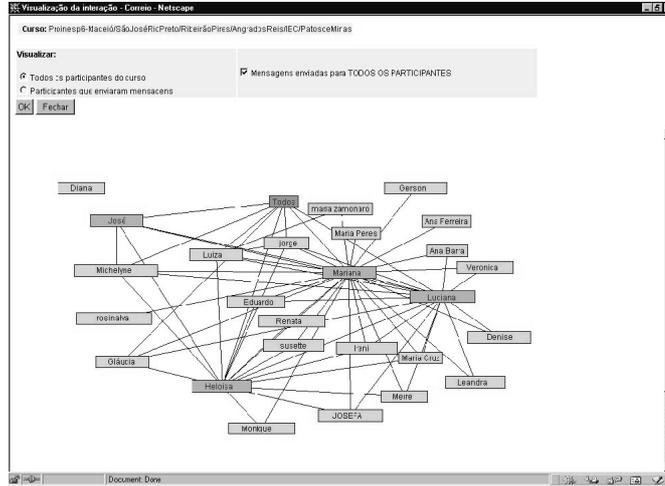


Figura 4: Visualização da Interação

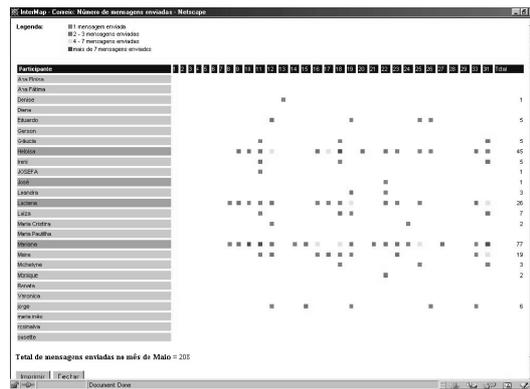


Figura 5: Representação bi-dimensional com código de cores

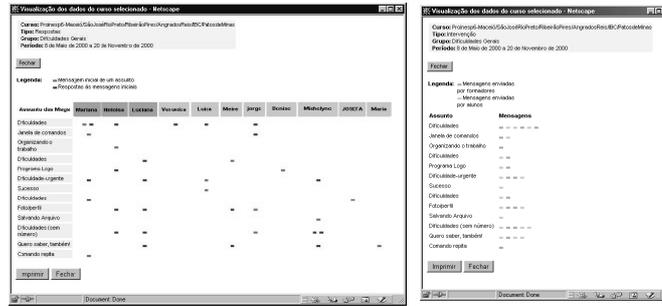


Figura 6.: Representação bi-dimensional (Estrutura de Respostas e Intervenção do Professor)

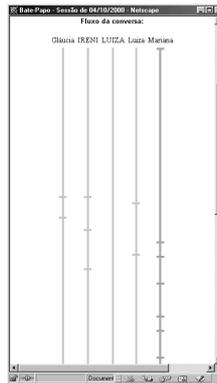


Figura 7: Visualização de uma sessão de bate-papo (trocas de turno e pausas)

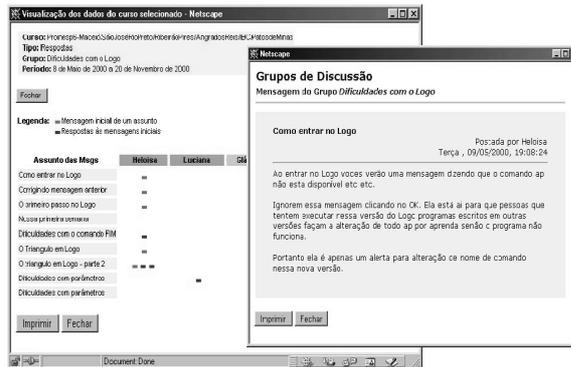


Figura 8: Janela com detalhes do conteúdo de uma mensagem

Personal and group spaces: Integrating resources for users of digital libraries

J. Alfredo Sánchez, Carlos Proal, Dámaris Pérez, Ana R. Carballo

Library and Center for Research in Information and Automation Technologies of the
Universidad de las Américas-Puebla
A.P. 100 UDLA-P Cholula, Pue. 72820 México
{alfredo, carlos, sp098867, carballo}@mail.udlap.mx

Abstract. *This paper presents an approach to integrate highly diverse resources available in digital libraries via personalizable interfaces and virtual collaboration areas. Digital libraries comprise vast digital repositories and a wide range of services, user environments and interfaces, all intended to support learning and collaborative research activities. We are developing a distributed digital library for which services and interfaces include, for example, mobile agents for federated information retrieval, recommendation agents, 3D visualization aids, and access to digital documents that support specific courses and projects. The diversity of these interfaces, plus the volume and dynamism of the digital library's underlying collections, result in a complexity that has the potential to make the digital library unwieldy for the user. In order to assist users in dealing with this complexity, we have designed environments, termed personal and group spaces, which provide users with means to access and control all available resources in a uniform fashion from a single vantage point. We discuss our design and development experiences as well as initial usage results.*

Keywords: *personalized interfaces, collaborative interfaces, digital libraries, personal spaces, group spaces, agents.*

1. Introduction.

The systems we have come to know as “digital libraries” are not really libraries in the usual sense. The abstractions of *content* and *services* that generally describe physical libraries have been used by researchers to convey some of the functionality enabled by digital libraries, but this metaphor does not even suggest the major differences that the introduction of a new medium entails. In addition to a new substrate, digital libraries comprise new information units and genres, allow for multiple novel organization schemes, and make diverse browsing and searching mechanisms possible. Users of digital libraries require new skills, new tools and new interfaces to cope with the complexity of such a diverse system and to fully exploit its potential.

This paper analyzes some of the major problems faced by users when using collections and services provided by digital libraries and presents an approach for the integration of digital library resources which is based on the concepts of *personal* and *collaboration spaces*. We posit that users should have at their disposal means to create virtual areas within the digital library in which they (or their agents) can place information objects that are relevant to their interests and ongoing tasks. We refer to this sort of virtual place as a personal space. Additionally, digital libraries should also provide users (usually working remotely) with

virtual places where they can meet to discuss objects and topics of interest and to work collaboratively on group projects while maintaining the capability of accessing materials in digital collections. We refer to this second category of virtual place as a group space.

Our group has been conducting research in the area of digital libraries and has produced advances both in the construction of digital contents and in the provision of services for users in the context of a distributed system architecture. We have implemented and deployed various architectural components which are available to a wide user community. In this context, we have designed and implemented two versions of personal spaces and one of group spaces. We report our design and development experiences in this particular area of our digital library.

The remainder of the paper is organized as follows: Section 2 elaborates on our digital libraries effort focusing on human-computer interaction issues, architectural components and interfaces and services to be integrated. Next, we present the design rationale for personal and group spaces in Section 3. We discuss prototypical implementations for our design in Section 4 and qualitative evaluations and usage results Section 5. We briefly review projects elsewhere that are related to our work in Section 6. Finally, we discuss the status of our project as well as ongoing and future work in Section 7.

2. Digital Libraries Context

Our group has developed a system architecture for a digital library that addresses the needs for information management, communication and collaboration among a highly distributed community of users [Sánchez and Leggett 1997]. We have aimed to build both a practical, functional digital library and a testbed for research of open issues in the field, including aspects of personalized and group interfaces, collaborative work and information visualization. We refer to our digital libraries initiative as “University Digital Libraries for All”, or U-DL-A.

2.1 U-DL-A components.

The realization of our digital library has called for research and development efforts in three major areas: building content, designing components to provide general infrastructure (such as information retrieval services, multimedia management and agent essentials) and user interfaces and environments. With regard to digital contents, we have made progress in the construction of collections of digital theses, special collections from our library and university publications. Our advances in the area of general infrastructure include a distributed framework that integrates services and interfaces, with implementations using KQML, CORBA and RMI [Barceinas et al. 1998; Coccoletzi 2001] as well as components implementing various popular information retrieval mechanisms. Finally, in the area of user interfaces and collaboration environments, our work has resulted in various access and visualization mechanisms [Sánchez 2001]. Given the strong relationship of these components with the work presented in this paper, we provide some additional details about them in this section.

UVA. U-DL-A Visualization Aid (UVA) is a component of our digital library that allows users to visualize large information spaces that are organized hierarchically as 3D trees [Proal et al. 2000]. UVA has been tested and has received good reviews when used with

collections that classified according to the schema of the US Library of Congress and other domain-specific taxonomies.

TD (Digital Theses). Our library is now providing means to store and retrieve the theses produced by the university's graduating students. We offer guidelines for students to produce TD-compliant digital documents, mechanisms to convert from various word processing formats into our TD data model, and a wide range of access and retrieval options.

Zeus. We also have developed an environment for collaborative revision and annotation of digital theses [Fernández et al. 2000a]. This is an asynchronous, role-aware interface, highly personalizable system component that allows theses advisors and committee members to review and annotate digital theses via user-defined conventions.

Viajerus. We are collaborating with other digital library efforts in different institutions. In order for our users to benefit from this collaboration, we have devised mobile agents that attend to user queries and visit digital collections at our partner's libraries and gather information in a transparent fashion [Chevalier et al. 2001].

SyReX. Users of U-DL-A's digital library can receive content-based and collaboratively generated recommendations about relevant information resources via this software component [Ramírez 2001].

RDU (U-DL-A Digital Reserve). This component provides access to digitized materials that are traditionally available at the Reserve section of academic libraries. Users have priority access to materials made available for their courses or research projects. Materials may be viewed on graphical computer displays and searched over according to various criteria [Rodríguez 2001].

2.2 Aiming for seamlessness

One distinctive element of our work in digital libraries is our view of information and collaboration spaces as comprising both the digital and physical realms. We believe the needs posed by scholarly activities can be best satisfied if traditional resources as well as the novel digital facilities are accessible to users at any given time in a seamless fashion. In this regard, we have been integrating traditional on-line catalogs and user services (such as lending materials, selective information dissemination and other notification services) into our new developments. As can be expected, software components such as UVA, SyReX and RDU consider that both digital and physical resources should be accessible to users.

Even though each software component may facilitate seamless access to some library resources, the diversity of such components as well as the size and dynamic nature of digital collections and user communities have the potential to make the digital library a complex and unwieldy environment for end users. Means are needed to ensure that individual users have access to exactly the resources they are entitled to according to their roles, needs and preferences. Providing more than what a user needs unnecessarily increases the system's complexity and the user's cognitive overload. On the other hand, limiting access excessively would result in system under-utilization (and user frustration). The needed compromise could be the result of explicitly eliciting and addressing every user's needs and preferences. Alternatively (and this is our approach), users could have at

their disposal interfaces which allow them to define and gradually refine the collections, portions of collections and services that best suit their needs.

Digital library users typically need to work collaboratively. More often than not, a given user belongs to more than one regular group of users. It is also possible that library resources are disjoint for every group a user belongs to, or that some materials need to be used in several work groups. Thus, for example, a user may participate in a class discussion for which she is the instructor or facilitator and also in a technical committee that is reviewing a paper submitted to a conference. For her class, she may suggest a reading (available at the digital library) which happens to be, say, a missing reference in the paper being reviewed by the other group in which she participates.

The work described in this paper aims to advance along these lines: facilitating the creation and maintenance of personalized areas (which we call personal spaces) that integrate resources relevant for a given user, and making it possible for users to seamlessly move from personal to work group areas and between various group areas (which we call group spaces).

3. Personal and group spaces.

In the context of a digital library, we define a *personal space* as a virtual area that is generated, owned and maintained by a user to persistently keep resources (objects, agents, etc.) or references to resources which are relevant to a task or set of tasks the user needs to perform in the digital library. Personal spaces may thus contain digital documents in multiple media, personal schedules, visualization tools, and user agents that have been delegated various tasks. Resources within personal spaces can be pre-assigned according to the user's role. For example, a graduate student would have access to course-specific reserve materials, visualization tools and interfaces to upload thesis chapters for revision by a committee. Agents may be available for recommending library materials that are relevant to the thesis and the personal space could be enriched by the agent's suggestions.

Similarly, we define a *group space* as a virtual area in which library users can meet to conduct collaborative activities synchronously or asynchronously. *Explicit* group spaces are created dynamically by a group leader or facilitator who becomes the owner of the space and defines who the participants will be. *Implicit* group spaces can be generated automatically when a number of users have been detected to have similar user profiles or interests around a given topic or task. In addition to direct user-to-user communication, users should be able to access library materials and make annotations on them for every other group participant to see.

Ideally, users should be able to move seamlessly (and carry digital library materials with them) between personal and group spaces or among group spaces to which they belong. Figure 1 illustrates a system with four users (each with one personal space, P1-P4) and two group spaces (G1, G2). Each personal space includes a number of library resources (Ri's), some also included as part of one or more group spaces. For example, resources R4, R5 and R8 in personal space P2 have been carried to group space G1 to be shared with the user who owns P1. Moreover, resource R8 is also being shared through group space G2 with the users who own personal spaces P3 and P4. It can be noted also that all resources in personal space P4 (R12 and R13) are being shared with all other users via group space G2.

We can also regard this scenario as involving only three users, one of which has defined two personal spaces (say P3 and P4) to handle resources R9-R13 and sharing some of the resources in P3 and all resources in P4 with the other users via the G2 group space.

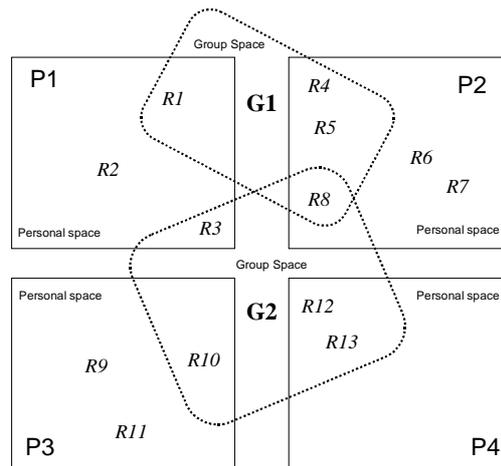


Figure 1. Four personal spaces and two group spaces
(P_i = Personal Space; G_i = Group Space; R_i = Library resource).

For the sake of clarity, in Figure 1 we have considered only disjoint resource sets for each personal and group space. It may also be the case, however, that a given resource is referenced in multiple personal or group spaces.

Basic functionality required for personal spaces includes the capabilities for viewing, launching and monitoring library services, agents and applications. As for group spaces, they should provide users with means to easily become aware of users and resources that are present in a given group space at any time, as well as mechanisms to communicate with other users and make annotations on library resources. Naturally, there should also be means to move from personal to group spaces and to carry materials from

4. Implementation

We have produced prototypical implementations of the personal and group spaces concepts to provide a homogeneous interface to the various user services and environments described in Section 2.1. We based our designs on the well-known *room* metaphor [Greenberg and Roseman 1998; Henderson et al. 1986]. Personal spaces are thus conceived as rooms that every user may configure by defining physical characteristics (color, furniture, layout) and by adding library resources that are used frequently or support user tasks. We also have emphasized the role of every user in our interface design [Shneiderman and Plaisant 1994]. When users register to access the digital library, they specify the role(s) they will be playing in the system. Typically, a user selects or is assigned a combination of the various existing roles: graduate or undergraduate student,

researcher, full-time or part-time professor, librarian, administrative staff, or system administrator. The initial room configuration is determined by the user's role(s).



Figure 2. The interface of MiBiblio

4.1 MiBiblio

Our first prototypical implementation of personal spaces, which we termed *MiBiblio* [Fernández et al. 2000b], was made available to users as Java applets that could be downloaded and run at the client level. Figure 2 illustrates the interface of MiBiblio. The idea is straightforward: Users may configure their personal space by changing a room's physical characteristics and placing book icons (representing library resources) into bookshelves that can be added and moved around at will. Drag-and-drop functionality can be applied practically to any element in the interface. The figure also shows icons for the mobile search agents (owls) and recommender and reference librarian agents (animistic books). The detailed functionality of these agents is beyond the scope of this paper but is available in related publications [Chevalier et al. 2001; Ramírez 2001; Sánchez et al. 2001]. Pull-down menus include options for adding or removing elements from a room, associating library resources with icons, and adding or removing rooms from a personal space. New elements can be added by specific user actions or as a result of agent recommendations or search processes.

4.2 MiStudio

Our primary target community of users includes some 8,000 students, professors, researchers and librarians and administrative staff at our university. Given that not all the available computer facilities (mostly SunRay clients and Macintosh personal computers) necessarily comply with the requirements imposed by MiBiblio, we developed a "lighter" version of the interface, which we termed MiStudio.

Figure 3 shows a typical configuration of the MiStudio interface. Although the graphics used for this representation are of higher quality, the elements in the interface are pre-defined and the user can only modify the associations between specific objects in the room and library resources. MiStudio is based on dynamic HTML pages, which implement the functionality required to launch applications as needed by the user. In the figure, for example, the picture of Porfirio Diaz on the table is linked to the digital collection of telegrams generated during the administration of this President of Mexico (from 1876 to 1911), whereas the envelope on the table is linked to the Digital Reserve System (RDU) which provides access to documents in the library's reserve section.



Figure 3. The Interface of MiStudio

4.3 EGA

We also have implemented a prototype for group spaces which we refer to as *EGA* (Spanish acronym for Learning Group Spaces). Our current version of EGA implements explicit group spaces (as per our definition in Section 3): any user of the digital library may create (and subsequently own) a group space and send invitations to users who are members of a team or who are potentially interested in a particular discussion subject.

Figure 4 illustrates a session of five users who have started a discussion in EGA. Upon entering the "room" representing the group space, every user sees a picture of himself, images of the other users currently logged in, as well as icons that represent library resources (in the bookshelves at the bottom), which have been brought to the room to support the discussion. Users participate in the discussion by posting their comments via a text interface and pointing to digital library resources. Users can make annotations on shared materials and see other users' annotations. The entire discussion can be saved so latecomers may join a meeting and get up to speed quickly, or the discussion may be suspended and retaken by the group at a different time.

In EGA, users can drag their own images and drop them near another user, group of users or library resources. This is intended to reflect positions in a discussion in a visual fashion that can be perceived by all group participants. In this regard, images of users are distributed evenly on the display as users sign in. When an issue is raised, participants may literally take sides: If they agree with a position, they move their images near the user with whom they agree. Additionally, new rooms (or sessions) may be created to include different sets of users for addressing sub-topics or new discussion topics.

5. Evaluation

Our implementations of personal and group spaces have undergone various preliminary usability and performance tests with encouraging results. We have focused initial tests to our main target communities: college students, professors and administrative staff. Test participants have included over 20 library personnel, ten faculty members, 20 undergraduate students and five graduate students. For our tests we have used mainly our collection of digital theses and the online catalog of our physical library, which comprises over 230,000 physical volumes.

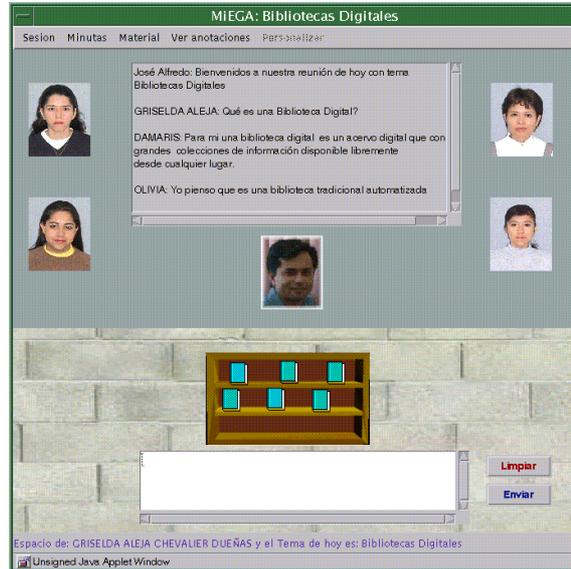


Figure 4. The interface of EGA – Learning Group Spaces.

In addition to usability inspection tests, we have been observing users experiment with the prototypes. With only minimal initial guidance, users of MiBiblio have been able to personalize their information space by creating rooms, changing their appearance, and more importantly, by gathering library materials (mostly theses) and references (online catalog entries) and placing them in bookshelves. From their personal space, users also have been able to open digital documents and launch applications such as UVA and Zeus, and to generate and monitor instances of the SyReX recommender agent. MiBiblio has been in use for about nine months. However, some of the services and applications described in this paper have become available more recently.

As stated earlier, MiStudio was developed in response to infrastructure limitations to support the tools used in MiBiblio, as many of our target users have Macintosh computers and the required version of the Java plug-in is not available for this platform as of this writing. Users who have tried both alternatives have preferred MiBiblio's highly personalizable functionality, but have rated MiStudio favorably with regard to its performance. MiStudio has been available for about only one month and hence a more formal comparative evaluation has yet to be undertaken.

As for EGA, we have observed how it has been used formally as a means for supporting a graduate course and also as an environment for the discussion of project ideas. Users have been able to start new group spaces easily and to join existing groups or ongoing discussions without any significant problems. Minutes (saved discussions) have been used extensively to record discussion results or just as a mechanism to suspend collaborative work to be resumed at a later time. Digital theses and book references have been carried to group spaces and users have been able to make annotations and examine their contents when suggested by other group participants.

In general, users of MiBiblio, MiStudio and EGA have found the interfaces attractive and intuitive. In response to explicit surveys, over 80 percent of the users considered personal and group spaces useful and a unifying view of library resources. In all cases, close to 70 percent of the users were able to carry out the tasks they were asked or intending to do.

6. Related Work

There is a significant amount of work that has nurtured, or has been developed in parallel to, our notions of personal and group spaces. In the area of personalization of information spaces, DLITE [Cousins et al. 1997] introduces the concept of personal *workcenters*, in which users can perform tasks and access distributed information in a transparent manner. Among the differences with our work we can mention our emphasis on the integration of physical and digital resources and the incorporation of agent representations in personal spaces.

The impact of “one-stop” library portals on libraries, information providers and users, from the organizational, cultural and learning perspectives is discussed at length by Lakos and Gray [2000]. An example of library portal is Cornell University Library’s *MyLibrary*. This project comprises two main services: MyLinks, a tool for collecting and organizing resources for private use by a patron, and MyUpdates, a tool to help scholars stay informed of new resources provided by the library. This is an important effort in personalizing a commercial online catalog. Our emphases on digital contents, graphical interface components and agent-based interaction are distinguishing aspects of our work with respect to Cornell’s.

An architectural proposal for personalized information environments that is more oriented to digital libraries is presented by French and Viles [1999]. They consider personalized collections built from centrally maintained indices or indices constructed by a client from distributed repositories. In our architecture, we also consider the participation of mobile agents that travel through a federation of digital libraries so users may transparently access distributed repositories from a uniform interface.

Work aimed at supporting group activities is abundant in the CSCW/groupware literature. Our concept of group spaces particularly has built upon advances in group awareness, annotation, and synchronous learning spaces. In particular, the awareness mechanisms in our implementation of group spaces have been used before in early collaborative environments such as the Grove [Ellis et al. 1991] and Sasse [Baecker et al.1993] group editors.

7. Ongoing and future work

As our digital library work becomes more widely used both within our community and among external users, we are continually receiving feedback, adapting our software development plans and finding new application areas. We are currently working on a better integration of personal and group spaces. In upcoming versions of MiBiblio and MiStudio, "doors" will appear in rooms representing personal spaces for each of the available group spaces. Users will be able to exit a personal space to enter a specific group space by traversing the appropriate door. We have defined various formal tests of the prototypes, particularly in the areas of usability, robustness and scalability. We are currently in the process of applying these tests.

One feature of EGA that has not been experimented with more fully is the capability of group participants to move their individual images around the group space to create user clusters that reflect positions in a discussion. We intend to encourage users to take advantage of this facility and report results in the near future. We also plan to explore the definition of implicit group spaces to supplement EGA's current functionality. Our own previous work in a related project [Sánchez et al. 1998] can be adapted to our ongoing digital libraries initiative. In this area, we are currently exploring the use of videoconferencing as a way to widen the relatively limited communication channel provided by our text-based conversational component. Finally, we are working on the definition of an appropriate upper limit for the number of participants in group spaces, as in some cases problems have been observed as the number of users grows.

From our results thus far, we believe personal and group spaces are a promising approach to the integration of highly distributed, heterogeneous information spaces. Our design and prototypical implementations show that our approach may effectively help users in coping with the size, complexity and dynamism of digital libraries.

Acknowledgments

This work has been supported in part by grants from the Mexican National Council for Science and Technology (Conacyt, project 35804-A) and its Network for the Development and Research in Informatics (REDII).

References

- Baecker, R.M., Nastos, D., Posner, I.R., and Mawby, K.L. 1993. The user-centered iterative design of collaborative writing software. In *Proceedings of INTERCHI '93*, 399-405.
- Barceinas, A., Sánchez, J. A., Schnase, J. L. 1998. MICK: A KQML inter-agent communication framework in a digital library. In *Memorias del Simposium Internacional de Computación* (CIC, México City, Nov.), 66-79.
- Chevalier, G., Sánchez, J. A., Fernández, L., Nava, S. 2001. A framework based on mobile agents for distributed information retrieval. *Proceedings of the Third Mexican International Conference on Computer Science (ENC 01, Aguascalientes, Mexico, Sept.)*. *Forthcoming*.
- Cohen S., Ferreira J, Horne A., Kibbee B., Mistlebauer H., Smith A. 2000. MyLibrary: Personalized electronic services in the Cornell University Library. *D-Lib Magazine* 6, 4 (April).

- Cousins, S., Paepcke, A., Winograd, T., Bier, E. A., Pier, K. 1997. The digital library integrated task environment (DLITE). In *Proceedings of the 2nd ACM International Conference on Digital Libraries (DL'97, Philadelphia, Pa., July)*, 142-151.
- Cocolezzi, H. 2001. SerGen: Servicios Generales... B.Sc. Thesis. Department of Computer Systems Engineering. Universidad de las Américas-Puebla, AP 100 UDLA-P, Cholula, Puebla 72820, México (May). Also available from the Digital Theses Collection at <http://biblio.udlap.mx/tesis>.
- Ellis, C.A., Gibbs, S.J., & Rein, G.L. 1991. Groupware: Some issues and experiences. *Communications of the ACM* 34, 1, 38-58.
- Fernández, L., Sánchez, J. A., Flores, L. 2000a. An environment for the collaborative revision of digital theses. In *Proceedings of the Sixth International Workshop on Groupware (CRIWG 2000, Madeira Island, Portugal, October 18-20)*. IEEE Computer Society, Los Alamitos, Calif.
- Fernández, L., Sánchez, J. A., García, A. 2000b. MiBiblio: Personal Spaces in a Digital Library Universe. In *Proceedings of the Fifth International ACM Conference on Digital Libraries (DL'00, San Antonio, Texas)*. 232-233.
- Greenberg S. and Roseman, M. 1998. Using a room metaphor to ease transitions in groupware, Research report 98/611/02, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, January. <http://cpsc.ucalgary.ca/grouplab/papers,1998..>
- Henderson, D. Austin, and Card, Stuart. 1986. Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics* 5, 3, 211-243 .
- Lakos A., Gray C. 2000. Personalized library portals as an organizational culture change agent: Reflections on possibilities and challenges. *Information Technology and Libraries* 19, 4 (December), 169-174.
- Proal, C., Sánchez, J. A., Fernández, L. 2000. UVA: 3D representations for visualizing digital collections. In *Proceedings of the 3rd International Conference on Visual Computing (Visual 2000, Mexico City, Sep.)*, 185-192.
- Ramírez, M. 2001. SyReX: Recomendaciones... B.Sc. Thesis. Department of Computer Systems Engineering. Universidad de las Américas-Puebla, AP 100 UDLA-P, Cholula, Puebla 72820, México (May). Also available from the Digital Theses Collection at <http://biblio.udlap.mx/tesis>.
- Rodríguez, J. 2001. RDU: Reserva digital de U-DL-A. B.Sc. Thesis. Department of Computer Systems Engineering. Universidad de las Américas-Puebla, AP 100 UDLA-P, Cholula, Puebla 72820, México (May). Also available from the Digital Theses Collection at <http://biblio.udlap.mx/tesis>.
- Sánchez, J. A. 2001. HCI and CSCW in the context of digital libraries. *ACM Conference on Human Factors in Computing Systems (Extended Abstracts, CHI 2001, Seattle, Wash.)*
- Sánchez, J. A., Leggett, J. L. 1997. Agent services for users of digital libraries. *Journal of Networks and Computer Applications* 21, 1, 45-58.

- Sánchez, J. A., García, A. J., Proal, C., Fernández, L. 2001. Enabling the social construction and reuse of knowledge through a virtual reference environment. In Proceedings of the 7th International Workshop on Groupware (CRIWG 2001, Darmstadt, Germany, Sept.). *Forthcoming*.
- Sánchez, J. A., Fernández, L., and Schnase, J. L. 1998. Agora: Enhancing awareness and collaboration in floristic digital libraries. In *Proceedings of the Fourth International Workshop on Groupware (CRIWG'98, Rio de Janeiro)*. 85-95.
- Shneiderman B., Plaisant, C. 1994. The future of graphic user interfaces: personal role managers UMD CS-TR-3285, CAR-TR-713, ISR-TR-94-48 People and Computers IX, British Computer Society's HCI 94 (Glasgow, Scotland, Aug.) CU Press (Cambridge, U.K.) 3-8.

Personalizando a Interação Humano Computador para Buscas em Bibliotecas digitais

Profa. Msc. Jiani Cordeiro Cardoso¹, Prof. Dr. João Batista Oliveira²

¹PUCRS Campus Uruguaiana, BR 472 – Km 7
97500-970 - Uruguaiana, RS

²PUCRS – Porto Alegre, Av. Ipiranga, 6681 - prédio 16
90619-900 – Porto Alegre, RS

nani@pucrs.campus2.br, oliveira@inf.pucrs.br

Resumo. Este artigo apresenta a proposta de uma interface de consulta, personalizável, para uma Biblioteca Digital. A interface, gerada a partir da seleção de estilos de interface e opções de busca disponíveis, é configurada de acordo com as preferências dos usuários e monitorada pelo sistema. As opções de busca, selecionadas na interface adaptada ao perfil do usuário, poderão ser salvas, para poderem ser utilizadas em uma outra consulta ou acesso.

1. Introdução

A Internet evolui em um ritmo cada vez mais acelerado. Diretórios, endereços e sites desaparecem à mesma velocidade com que são criados. Existe um volume imensurável de informações, e como consequência uma facilidade dos usuários para se “encontrarem” ou se “perderem” neste universo virtual.

Enquanto a forma de distribuição das informações na WWW pouco se alterou nos últimos anos, a tecnologia para a apresentação destas informações teve uma evolução rápida e fértil. Inicialmente baseada no paradigma de hipertextos, hoje convivem estilos tão diferentes de interação quanto a realidade virtual não imersiva (VRML), aplicações com interfaces CGI (*Common Gateway Interface*) e documentos interativos (HTML, CSS (*Cascade Style Sheets*) e Javascript, entre outros [WIN 99], sendo que elas podem atuar em conjunto em um mesmo projeto.

O surgimento da WWW representa mais um desafio para o desenvolvimento de interfaces, uma vez que, as atualizações podem ser diárias, tanto ao nível de conteúdo como de funcionalidades apresentadas e que podem ser modificadas rapidamente.

Embora muitos avanços em termos de tecnologias aplicadas às interfaces estejam ocorrendo, se considerarmos os Sistemas de Recuperação de Informações, principalmente sites de busca e bibliotecas digitais (BDi), podemos concluir que eles continuam a usar índices que são muito similares aos utilizados pelos bibliotecários há anos atrás. O que mudou então? [BAE 99].

Três mudanças fundamentais têm ocorrido devido aos avanços das modernas tecnologias de computação e a explosão da Web. Primeira: tornou-se mais barato ter acesso a várias fontes de informação, e isto, permitiu alcançar uma grande audiência se comparada as demais possibilidades, tais como assinatura de periódicos, deslocamento físico até os centros de informação, etc. Segunda: os avanços em todos os tipos de comunicação digital

forneceram um acesso maior às redes. Isto implica que a informação está sendo disponibilizada mesmo que distante fisicamente. Terceira: a liberdade de divulgar qualquer informação por qualquer pessoa que a julgue útil tem contribuído para o aumento de popularidade da Web. Pela primeira vez na história, muitas pessoas têm acesso livre a uma grande e nova forma de publicação [BAE 99].

Desta forma, as BDi têm sido consideradas por vários pesquisadores como uma saída viável para a organização deste espaço de informação disponível atualmente pelas redes de computadores [STR 96][BIR 94] porque se apresentam como uma alternativa para ampliar as condições de busca, disponibilidade e recuperação de informações de maneira globalizada, qualitativa e pertinente [MAR 97]. Associações científicas e universidades ligadas ao contexto de educação a distância têm reconhecido a necessidade do desenvolvimento de acervos digitais provendo organização e transparência no acesso.

Porém, as BDi têm enfrentado um desafio em sua concepção que é a criação de interfaces que auxiliem o usuário a lidar com o volume, complexidade e dinamismo dos atuais e futuros repositórios de informação digital. Um dos desafios neste momento é encorajar os usuários a explorar as potencialidades presentes nessas interfaces, uma vez que, a diversidade de apresentação e o grau de conhecimento dos usuários sobre sua utilização têm dificultado a aprendizagem e conseqüentemente limitado a sua utilidade.

2. A Biblioteca Digital como suporte a educação a distância

Embora as BDi tenham surgido de uma evolução da Biblioteca Tradicional, elas têm sido amplamente discutida no contexto de suporte a Educação a distância.

A educação a distância é uma modalidade de ensino que promove novas oportunidades de ensino e aprendizagem, que tem sido consideradas positivamente durante este século. Conhecida como educação alternativa ou não formal, os programas e cursos oferecidos através do ensino a distância pelas instituições estão voltados tanto para o autodesenvolvimento como para a educação continuada profissionalizante [BLA 98].

Para (Tiffin & Rajasingham *apud* [BLA 98]) no aprendizado a distância é especialmente relevante o acesso às bibliotecas. Além das aulas assistidas em tempo real é preciso elaborar e distribuir aos estudantes materiais suplementares variados. Estes autores complementam que quando for possível realizar o *download* de determinados textos de livros em determinadas bibliotecas, a classe virtual terá sua biblioteca virtual. Se isto era utópico há poucos anos atrás, hoje podemos observar inúmeras iniciativas no mundo inteiro para disponibilizar acervos digitalizados em bibliotecas de acesso globalizado proporcionado principalmente através da Internet.

As BDi podem ser utilizadas como centro de material suplementar. Uma vez que cursos a distância sejam oferecidos, também novas opções de coleta de informações precisam ser disponibilizadas para a realização de seus trabalhos. Elas podem auxiliar educandos a distância a localizar material específico; auxiliar os usuários a identificar recursos próximos a sua residência; desenvolver estratégias de pesquisa com os aprendizes [BLA 98]. É importante ressaltar que estes recursos *on-line* devem ser desenvolvidos de modo a focar as necessidades específicas de cada grupo de usuários.

Desta forma, esse novo acervo vai permitir que sejam eliminadas, de fato, as paredes da sala de aula, sendo o aprendizado para os alunos virtuais realizado independentemente de sua distância ou localização [CUN 00].

2.1 A problemática da interação humano computador em Bibliotecas Digitais

A busca de informação é um processo impreciso. Quando determinado usuário acessa um SRI (Sistema de Recuperação de Informação) ele freqüentemente tem somente uma compreensão confusa de como pode alcançar seu objetivo. Deste modo, a interface de usuário deveria ajudá-lo a compreender e expressar as necessidades de informação. Ela deveria também ajudá-lo a formular suas consultas, selecionar entre as fontes de informações disponíveis, compreender os resultados da busca e guardar o caminho do progresso de sua busca [BAE 99].

Isto ocorre porque em muitos casos a interface homem-computador é bem menos compreendida do que outros aspectos da recuperação de informações, em parte porque os humanos são mais complexos do que os sistemas de computadores e suas motivações e comportamentos são mais difíceis de medir e caracterizar [BAE 99].

Inúmeras tecnologias existem ou se encontram atualmente em desenvolvimento que podem ser aplicadas na concepção de tais interfaces: a navegação hipertexto que facilita a exploração de grandes espaços de informação [FOX 93]; técnicas de busca e recuperação de recursos de interesses específicos [BAE 99] [KOW 97]; técnicas de visualização do contexto e de vários níveis de detalhes da informação desejada [BAE 99]; além da integração do espaço e do tempo em ambientes de trabalhos computacionais [LEG 95]. Porém a exploração dessas técnicas ainda está em fase embrionária no atual cenário brasileiro. E ainda que a exploração dessas técnicas possa oferecer uma qualidade maior na interação entre o usuário e a informação, há uma série de outros fatores primários dificultando a compreensão das interfaces de BDi levando o usuário a fazer uso dos recursos mínimos oferecidos por estas. Dentre esses fatores, podemos destacar [CAR 99a]:

1. Muitos usuários não são especialistas em informática – e desta forma tem dificuldade para compreender as capacidades que lhe são oferecidas, como os operadores booleanos, busca exata, *wildcard*, por exemplo.
2. Diversidade de opções, sintaxe e terminologia – dificultando assim a memorização de ações a serem realizadas e que acabam inibindo o usuário a usá-las;
3. Frequentemente as interfaces de BDi apresentam uma “interface simples” e outra “avançada”, considerando o usuário iniciante ou avançado somente, não motivando ao descobrimento e utilização – quando o usuário se depara com formulários muito extensos há uma grande probabilidade dele acabar desistindo daquele site ou biblioteca.
4. Com muita freqüência os usuários alternam entre sites de busca até que um deles lhe apresente algo que chame sua atenção em relação ao que está sendo procurado; não há fidelidade por parte dos usuários, eles consideram mais interessantes tentar encontrar o termo especificado em outro *search engine* ao invés de tentar compreender como deve ser realizada a consulta naquele site.

5. Normalmente os *helps* apresentam todas as capacidades disponibilizadas num determinado site de busca ou biblioteca digital, porém são muito pouco atrativos passando despercebidos pelos usuários.

Esses fatores foram identificados a partir de um levantamento realizado junto a algumas BDi disponível na Internet, a fim de, identificar as estratégias de consultas apresentadas nessas bibliotecas e formas de auxílio oferecido aos usuários. Para o levantamento das capacidades de busca e propriedades, quatro BDi reconhecidas por suas comunidades científicas foram analisadas, a saber, *ACM Digital Library*, *New Zealand Digital Library*, *IEEE Computer Society Digital Library* e *American Memory Collections*. Embora existam vários projetos de BDi disponível na Internet, essas bibliotecas foram consideradas adequadas ao levantamento por encontrarem-se em um estágio considerável de disponibilização de materiais oferecendo opções diversificadas de busca.

É importante salientar que as diversas capacidades de busca oferecidas pelas BDi são extremamente importantes para o desenvolvimento e disponibilização de novas formas de acesso aos acervos capazes de facilitar a localização de itens relevantes com maior rapidez. Porém, é necessário explorar a interação nessas interfaces de maneira a preparar os usuários para utilizá-las, fazer com que o mesmo reconheça a importância e necessidade e saiba quando e como utilizar tais capacidades. As interfaces tanto para BDi como para sites de buscas precisam aumentar a clareza no controle do usuário sobre as ações, reduzindo, ao máximo, inconsistências.

3. A iLIB

A iLIB surgiu a partir da necessidade de explorar novas formas de interação capazes de auxiliar o usuário a compreender as capacidades que lhe são disponibilizadas, ao mesmo tempo, que tenta estimular o usuário a personalizar sua busca e utilizar as opções de refinamento da consulta que se apresentem disponíveis na interface.

A iLIB utiliza-se da estratégia de seleção de estilos de apresentação para as interfaces, a saber: interfaces básica, interface de menu e interface de seleção; que estejam mais próximas a sua compreensão ou que melhor se identifiquem com o seu perfil. Também são oferecidas, aos usuários, ajudas sensíveis à interface apresentada, destacando as características dos elementos que dela fazem parte e daquilo que o usuário possa realizar através da mesma.

Com este trabalho procuramos: facilitar a exploração das capacidades de busca; mostrar as alternativas disponíveis em cada estilo disponibilizado; auxiliar a motivar as características individuais dos usuários com a finalidade de reduzir a rejeição das interfaces e, por fim, oferecer uma interface personalizável.

Em sua íntegra, este trabalho, constitui a dissertação da autora que procurou identificar as problemáticas envolvidas no desenvolvimento de interfaces para SRI e desenvolver uma interface que explorasse algumas possibilidades de interação com a intenção de minimizar alguns dos problemas identificados e que pode ser consultado em [CAR 00].

3.1. Características da iLIB

Na iLIB a personalização da interface permite ao usuário alterar propriedades que lhe estiverem disponíveis de acordo com a sua preferência, fazendo com que o usuário possa sentir-se mais à vontade ao dispor dos elementos da maneira que melhor lhe satisfizer,

A arquitetura geral desse sistema foi inspirada na arquitetura de um modelo tradicional de um Sistema Tutor Inteligente (STI) apresentada em [GIR 97] onde se prevê a modelagem do aluno (usuário) na busca de uma instrução personalizada. As funcionalidades identificadas na arquitetura proposta e em uma arquitetura tradicional de um STI são: existência de uma *interface* (iLIB) que é o meio de interação entre o usuário e o sistema; uma *base de domínio* (opções para o usuário) onde estão mapeados os “conteúdos” a serem trabalhados. No caso da iLIB estes conteúdos são representados pelas opções de refinamento da consulta disponibilizados na iLIB; *um módulo de controle* responsável pelo gerenciamento das trocas e sincronia de atividades que ocorrem nos diferentes módulos e etapas da interação de modelagem do usuário a partir de suas preferências em termos de opções selecionadas.

Na modelagem desse protótipo foi considerado que uma vez selecionado um estilo de interface o sistema deve ser capaz de apresentá-lo, bem como as preferências definidas em termos de opções para os campos de pesquisa, coleções e ordenação dos resultados. A aquisição dessas informações pelo sistema irá ocorrer de forma explícita, ou seja, as preferências do usuário são baseadas nas escolhas feitas pelo mesmo na interface de busca as quais poderão ser disponibilizadas pré-selecionadas em uma próxima consulta ou acesso se o mesmo assim desejar. Para isto, o pesquisador precisará estar registrado no sistema como sócio e ser identificado através de uma senha para que a iLIB disponibilize a interface armazenada para aquele usuário.

A iLIB foi implementada utilizando HTML e Javascript para criação das interfaces de busca e para validação de algumas informações digitadas pelo usuário. Foram utilizadas também funções Javascript e CGI para criação de roteiros para o processamento dos formulários. Um protótipo executável baseado em diretivas foi desenvolvido para a realização de um experimento e testes com os usuários.

A iLIB buscou seguir os princípios da usabilidade, propostas por [NIE 93], através da exploração de regras e diretivas que pudessem ser aplicadas a proposta. Bem como, o desenvolvimento de um protótipo inicial que, após submetido a uma avaliação heurística com usuários e profissionais de áreas relacionadas, deu origem a iLIB apresentada a seguir. Desta forma, profissionais usuários de Sistemas de Recuperação de Informações de três áreas do conhecimento foram submetidos a entrevistas e questionários relativos a iLIB. Sendo assim foram consultados: 9 profissionais da Informática da PUCRS (2 bolsistas de iniciação científica, 6 alunos de mestrado e 1 aluno de doutorado, sendo este professor da universidade e membro da equipe do Laboratório de Biblioteca Digital do Campus Global); 6 Bibliotecários da Biblioteca Central da PUCRS que trabalham no Setor de disponibilização, consulta e suporte ao Sistema Bibliotecário; 5 Profissionais da Agência Experimental de Publicidade e Propaganda da PUCRS (2 Web designers e 3 Publicitários). Esses grupos de pessoas, formadas e atuantes em diferentes áreas, possibilitaram que suas opiniões pudessem ser consideradas tanto no escopo de usuários de SRI, na qual a iLIB está inserida, como no grupo de especialistas que devem fazer parte do projeto de interfaces. Embora houvesse outros grupos de pessoas que poderiam fazer parte da análise

da iLIB, devido a interdisciplinaridade desse trabalho, a seleção desses grupos foi dada a partir da disponibilidade dos mesmos em colaborar nesta etapa.

Primeiramente todos os grupos responderam a um questionário básico no qual foram perguntados sobre questões genéricas sobre os Sistemas de Buscas conhecidos, frequência de uso, características, etc. Através de um questionário específico sobre a iLIB os especialistas foram perguntados sobre preferências em termos de estilos apresentados, necessidade das opções disponíveis, adequação do *layout*, facilidades/dificuldades encontradas, sugestões de mudanças, entre outros. Esse questionário foi disponibilizado a cada integrante dos grupos envolvidos. Na coleta dessas informações, a autora deste trabalho realizou entrevistas informais com alguns membros dos grupos envolvidos a fim de discutir as opiniões e sugestões oferecidas no questionário.

Embora tenha sido submetido o mesmo questionário a todos, especial atenção foi dada às respostas e sugestões oferecidas pelo grupo da comunicação (designer e publicitários) em termos de *layout* (cores, ícones, terminologia) e navegação; ao grupo dos biblioteconomistas no que se referiu a distribuição das interfaces de consulta e adequação da interface, e ao grupo de informática sobre todos os aspectos de modo geral.

Os instrumentos de pesquisa elaborados foram construídos baseando-se em exemplos de instrumentos de avaliação de *software* desenvolvidos por especialistas. Os instrumentos retornados foram analisados e permitiram a identificação das alterações a serem realizadas na iLIB. A análise completa, bem como os questionários são apresentados em [CAR 00].

A Figura 1 apresenta a tela de acesso, responsável pela validação da senha do usuário que deverá reconhecê-lo e apresentar a iLIB da forma como foi definida por aquele indivíduo.

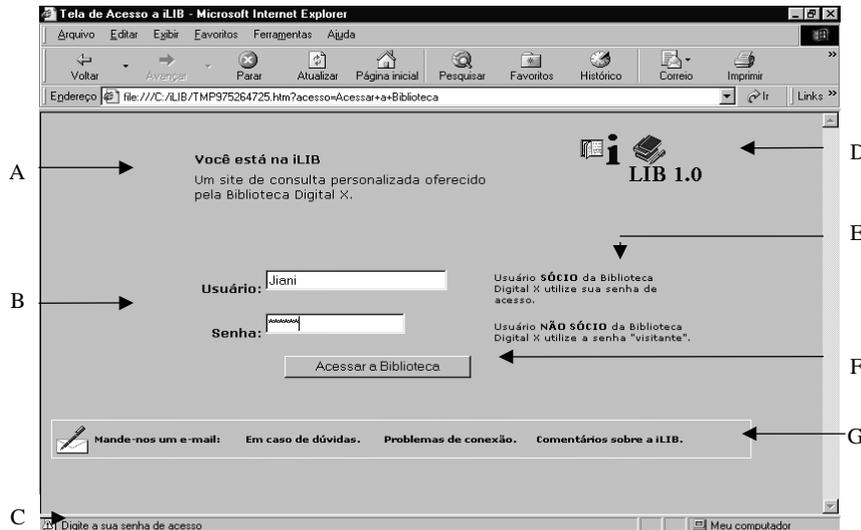


FIGURA 1 – Tela de acesso a iLIB.

Nela podemos destacar: A - uma breve apresentação da iLIB; B - Campos de entrada para nome e senha do usuário. C - Sugestões de ações utilizando como meio a barra de status. D - Ícone da iLIB, na qual é oferecida um *link highlight*¹ para informações sobre a iLIB. E - Informações sobre como deve ser o acesso a sócios e não sócios. F. Processo de validação da senha. G. *Links* para correspondência do usuário em caso de dúvidas, problemas de conexão e comentários.

A partir de uma interface principal o usuário poderá selecionar o tipo de interface que utilizará para realizar sua busca: interface básica, interface de menu ou interface de seleção. Ao clicar em interface básica a mesma é disponibilizada como é mostrada na Figura 2.

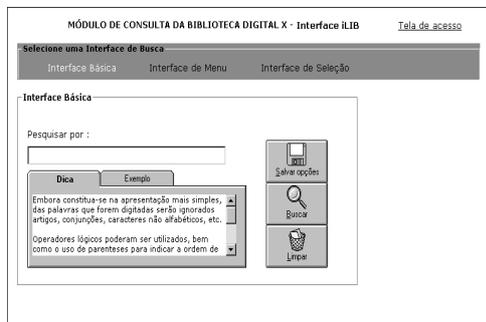


Figura 2 – Interface básica

A figura 2 apresenta a interface básica de consulta, da qual fazem parte dicas e exemplos de como expressar uma consulta utilizando-se dessa interface. As dicas poderão servir para melhorar a consulta do usuário ao informar que tipo de construção de consulta ele poderá utilizar, como ele deverá utilizar os operadores booleanos, por exemplo; já os exemplos servem para ilustrar as formas de apresentação de construção de consultas.

Se o usuário clicar sobre o botão “salvar opções” o sistema irá lhe informar que estará salvando esta interface como a de sua preferência e a lhe ser apresentada num acesso futuro como padrão.

Se o usuário clicar sobre a opção “buscar” o sistema deverá realizar a consulta. (É importante salientar que os mecanismos de busca não estão implementados, nosso propósito é explorar a personalização da interface apresentada ao usuário, logo não há atividade sob este botão). Caso o usuário clique sobre o botão limpar o sistema deverá limpar o(s) campo(s) preenchidos.

As figuras 3 e 4 apresentam a interface de menu e interface de seleção, respectivamente. Elas oferecem um conjunto de opções que possibilitam ao usuário refinar sua consulta, tais como seleção de uma coleção, tipo de organização dos resultados, seleção dos campos a serem pesquisados, entre outros. Nelas também são apresentados dicas e exemplos que podem auxiliar o usuário a utilizá-las de uma forma mais adequada. Basicamente as opções disponibilizadas nessas interfaces são as mesmas, porém apresentadas de forma diferente de modo a adequar-se ao perfil dos usuários. Alguns

¹ *highlight* – realça um termo ou frase, podendo ser um *link* para outros nodos.

usuários são mais orientados a menus, outros preferem ver rapidamente todas as opções disponíveis.

Da mesma forma que acontece na interface básica, o usuário ao clicar em “salvar opções” estará salvando esta interface como a sua interface personalizada, a sua iLIB, bem como os campos que ele tiver selecionado serão apresentados como padrão em uma nova consulta. A iLIB do usuário poderá ser alterada simplesmente ao salvar uma nova opção como padrão. O sistema então irá substituir a interface salva. Porém, cabe salientar que o usuário não é obrigado a salvar um estilo, ele pode simplesmente utilizá-los de acordo com a sua preferência. Também, cabe lembrar que para usuários não sócios a opção “salvar opção” irá se apresentar desabilitada.

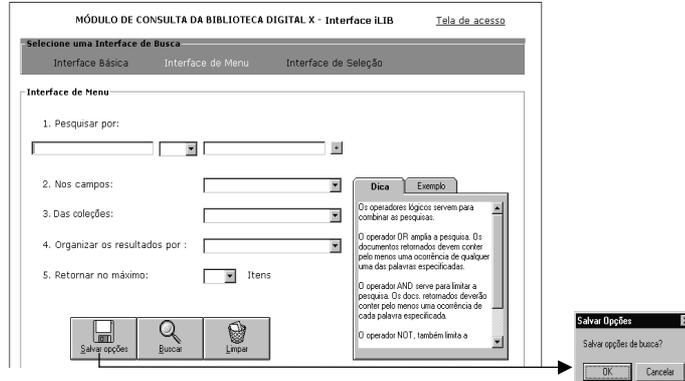


Figura 3 – Interface de Menu

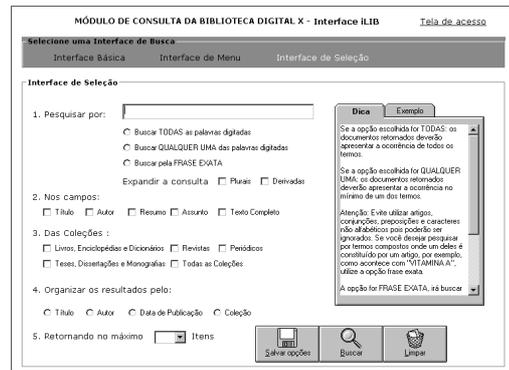


Figura 4 – Interface de Seleção

4. Iniciativas de personalização de Interfaces de BDi

Pesquisando projetos e *sites* de BDi que concentram estudos em interfaces de usuário encontramos alguns trabalhos recentes e interessantes que nos motivam a prosseguir as pesquisas em interfaces personalizadas para esse tipo de sistemas. A seguir, destacamos alguns destes apresentando suas características principais. Eles foram selecionados por apresentarem algum tipo de contribuição para a iLIB.

My library - é uma iniciativa da *Cornell University Library* para oferecer vários serviços personalizados aos estudantes, professores e equipe. Atualmente, ela consiste do *MyLinks*, implementado neste ano, que é uma ferramenta para colecionar e organizar informações para uso privado. De qualquer computador com conexão a Internet os seus utilizadores podem recuperar seus *bookmarks* contendo seus sites favoritos. Encontra-se em desenvolvimento *MyUpdates* é uma ferramenta para ajudar os estudantes a manterem-se informados de novos recursos disponibilizados pela biblioteca [COH 00].

Web Ártemis - é a interface de consulta da *University of Michigan Digital Library* que permite ao usuário criar um *bookmark* dos documentos que ela pode disponibilizar, os quais poderão ser recuperados através do *bookmark* criado. Para isso, o usuário deverá informar o título do documento, a URL do documento e o nome do autor [WEB 00].

A Interface de usuário da New Zealand Digital Library - a New Zealand Digital Library, da Nova Zelândia na Universidade de Waikato, desenvolveu um sistema de BDi pioneiro, com a promessa de modificar o modo das pessoas de encontrar informações na Internet. Ela oferece a seus usuários a possibilidade de realizar pesquisas em diferentes bases de dados bibliográficas disponíveis ao redor do mundo, através de uma interface comum de busca e recuperação da informação, com a intenção de disponibilizar formas de localização de maneira mais conveniente e confortável. Está disponível a qualquer usuário da Internet, sendo que a recuperação de documentos está condicionada a disponibilidade das coleções a que pertencem [NZL 00].

A Interface de usuário da Alexandria Digital Library - a Alexandria Digital Library oferece aos usuários registrados a possibilidade de selecionar suas áreas de interesse para a consulta utilizando as propriedades *panning and zooming*² sobre um mapa do mundo bidimensional [ALE 00].

O que todos esses trabalhos tem em comum com a iLIB é a intenção de satisfazer as necessidades de pesquisa dos usuários através da exploração de consultas personalizadas e/ou mais atrativas. Eles também apresentam o interesse dos grupos envolvidos em atender a diversidade de usuários em potencial de BDi.

5. Limitações e Trabalhos Futuros

Muitas vezes após ter planejado um trabalho surgem novos detalhes que até então não haviam sido reparados e que são capazes de alterar o planejamento inicial. Desta forma várias dificuldades se apresentaram quando começamos a implementar a iLIB:

² *Panning and zooming* - referem-se a ação de mover câmeras que podem ser esquadrihadas lateralmente entre cenários (*panning*) abrindo ou fechando para se ter maiores ou menores detalhes e conseguir uma visão mais ampla (*zooming*) [BAE 99].

Dificuldades para obter o conjunto de dados necessários – em relação a modelagem do usuário, vimos que a frequência de utilização do mesmo poderia ser baixa e levaria muito tempo até podermos disponibilizar uma interface personalizada levando em conta o número de acessos. Portanto, a personalização deveria ser oferecida desde o princípio.

Dificuldade em encontrar ícones adequados - Assim, os poucos ícones apresentados são utilizados para quebrar a linearidade das páginas e chamar a atenção dos usuários para pontos importantes, como a opção de “Salvar Opções” “Limpar”, porém não são metáforas bem associadas com elementos da vida real, como setas para direita, para esquerda, onde o processo de compreensão é bem claro. Semelhante problema também foi encontrado na escolha dos nomes das opções dos estilos no protótipo inicial, se estendendo também ao novo protótipo.

A iLIB não pertence a nenhum sistema já desenvolvido, ela constitui em um trabalho de pesquisa a ser ainda aplicado a uma interface de biblioteca. Levando em consideração que o presente trabalho constitui-se em uma proposta de personalização de uma interface, seria interessante integrá-lo a um sistema existente a fim de realizar novas experimentações com um grupo maior de usuários e, a partir disso, chegar a uma interface ideal. O experimento realizado inicialmente mostrou a grande necessidade de uma interação com os usuários.

A *layout* das telas poderia ser trabalho com especialistas das áreas de comunicação, Neurolinguística e Ciência da Informação. O domínio da iLIB poderia ser expandido de forma a oferecer outros serviços personalizados. E o uso do paradigma multiagente poderia ser analisado dentro da iLIB com a intenção de avisar aos usuários quando itens de interesse são adicionados ou atualizados na biblioteca, conforme a monitoração das coleções utilizadas por estes.

Outros aspectos de personalização como cores, *layout* da página de consulta, tamanho das letras, poderiam ser incorporados. A questão da apresentação e visualização de documentos encontrados também poderia ser expandida pela iLIB.

6. Conclusões

Ciente dos inúmeros desafios encontrados por usuários em interfaces de busca, tais como dificuldade em formular consultas, diversidade de opções de busca com comportamento e terminologia variada de um sistema para outro, *helps* extensos e pouco atrativos, vimos o desenvolvimento de interfaces personalizáveis como um passo em direção à resolução desses problemas.

Nesse sentido, este trabalho foi motivado pelas necessidades existentes de se desenvolver meios alternativos de interação homem-computador para que os usuários possam atingir vantagens efetivas no uso de BDi.

Desta forma, a proposta da iLIB concentrou-se em desenvolver formas personalizáveis de apresentação das opções disponíveis para uma busca, com o propósito de estimular as características individuais dos usuários e estimulá-lo ao descobrimento das possibilidades oferecidas, tendo como finalidade reduzir a rejeição das interfaces e minimizar algumas carências que atingem boa parte dos indivíduos que utilizam-se desses sistemas. De nada adianta oferecer mecanismos de busca avançados se suas características

não forem reconhecidas, é preciso explorar maneiras compreensíveis de auxiliar as pessoas a utilizar-se das propriedades disponibilizadas.

Para justificar nossa proposta, em partes deste trabalho, salientamos as fragilidades dessas interfaces, que tem sido foco de estudo principalmente nas áreas de IHC e Ciência da Informação. Reconhecemos que a utilização dos métodos de busca são extremamente úteis para a recuperação de itens relevantes a um determinado indivíduo e que a padronização das formas de apresentação de capacidades não garante a eficiência de uso, uma vez que os usuários precisam principalmente entender o processo de busca e saber utilizar as formas de refinamento.

Dado que o desenvolvimento de interfaces de usuário constitui uma área de pesquisa multidisciplinar que precisaria do envolvimento de vários profissionais inseridos no projeto [SHN 97], foram buscados subsídios através da literatura para aplicar da melhor forma possível na iLIB os elementos interdisciplinares intrínsecos ao trabalho, bem como uma vasta utilização de publicações das áreas de biblioteconomia e ciência da informação.

Este trabalho não tem a pretensão de ser uma solução a problemas identificados em interfaces de busca, nem tão pouco criar novas propriedades de consulta, mas indicar caminhos que precisam ser cada vez mais explorados para suprir tais deficiências, tais como a exploração de formas de auxílio, melhor apresentação das capacidades disponibilizadas, reconhecimento de usuários com diferentes perfis.

A iLIB enfatiza estratégias de busca existentes de maneira a apresentá-las ao indivíduos de uma forma mais participativa. Desta forma, os protótipos desenvolvidos são ferramenta de um estudo experimental que precisam ser continuados e aprimorados através da colaboração de equipes interdisciplinares e testes de avaliação.

Embora este estudo tenha sido desenvolvido como trabalho de mestrado no PPGCC ele será disponibilizado ao Laboratório de Bibliotecas Digitais da PUCRS para contribuir com as pesquisadas em desenvolvimento, neste centro, e para uma futura integração com as demais partes de um sistema digital, uma vez que a ele não foram incorporadas estruturas de banco de dados, mecanismos de indexação ou de busca de dados.

7. Referência Bibliográfica

- [ALE 00] **Alexandria Digital Library Project**. University of Califórnia, Santa Barbara. Manual do usuário. Capturado em Ago. 2000. Online. Disponível na Internet <http://fat-albert.alexandria.ucsb.edu:8827/tutorial.html>.
- [BAE 99] BAEZA-YATES, Ricardo, RIBEIRO, Berthier N. **Modern information retrieval**. New York, N.Y : Addison-Wesley, 1999. 513 p.
- [BIR 94] BIRMINGHAM et al. The University of Michigan digital library: This is not father's library. In: PROCEEDINGS OF DIGITAL LIBRARIES'94, 1994. **Proceedings...**Texas: Hypermedia Research Laboratory Texas A&M University, College Station, 1994. p.53-60.
- [BLA 98] BLATMANN, Ursula, DUTRA, SIGRID Karin Weis. **Atividades em bibliotecas colaborando com a Educação a distância**. 1998. Capturado em Ago. 2000. Online. Disponível na Internet <http://www.ced.ufsc.br/~ursula/papers/atividades-ead.html>.
- [CAR99] CARDOSO, Jiani. **Requisitos e funcionalidades em interfaces de Bibliotecas digitais**. Porto Alegre, 1999. 56f. Trabalho Individual I (Mestrado em Ciência da Computação) – Faculdade de Informática, PPGCC/PUCRS, set. 1999.

- [CAR 00] CARDOSO, Jiani. ILIB, uma proposta de interface de consulta personalizável para Bibliotecas Digitais. Porto Alegre, 2000. 112 f. Diss. (Mestrado) Faculdade de Informática, PPGCC/PUCRS, dez. 2000.
- [COH 00] COHEN, Suzanne et al. My library – Personalized electronic services in the Cornell University Library. **D-LIB Magazine**. vol.6 n.4. April 2000. Capturado em out. de 2000.. Online. Disponível na Internet
[Http://www.dlib.org/dlib/april00/mistlebauer/04mistlebauer.html](http://www.dlib.org/dlib/april00/mistlebauer/04mistlebauer.html)
- [CUN 00] CUNHA, Murilo Bastos da. Construindo o futuro: a biblioteca universitária brasileira em 2010. **Ciência da Informação**, Brasília, v.29, n.1, p. 71-89, Jan/Abr 2000. Capturado em nov. 2000. Online. Disponível na Internet <http://www.ibict.br/cionline/>
- [FOX 93] FOX et al. **Users, user interfaces, and objects: envision, a digital library**. Journal of the American Society for Information Science 44, 8., September 1993. p. 480-491
- [GIR 97] GIRAFFA, Lúcia, M., M. **Fundamentos de sistemas tutores inteligentes**. Porto Alegre, 1997. 69f. Material de Aula – Tópicos Avançados II – Inteligência Artificial aplicada à educação (Mestrado em Ciência da Computação) - Faculdade de Informática, PPGCC/PUCRS, 1999.
- [KOW 97] KOWALSKI, Gerald. **Information retrieval systems: theory and implementation**. Boston: Kluwer Academic, 1997. 282 p.
- [LEG 95] LEGGETT, J., Nürnberg, P., and Schneider, E. Principles of human-centered information systems. In: **Proceedings of the International Conference on Artificial Reality and Tele-Existence/Conference on Virtual Reality Software and Technology, (ICAT/VRST)** '95, p. 173-176, Japan, Nov. 1995.
- [MAR 97] MARCHIORI, Patrícia Z. Ciberteca ou biblioteca virtual: uma perspectiva de gerenciamento de recursos da informação. **Ciência da Informação**, Brasília, v. 26, n.2, p. 115-124. maio/ago. 1997 Online. Capturado em jul. 1998. Disponível na Internet <http://www.ibict.br/cionline/artigos/>
- [NIE 93] NIELSEN, Jakob. **Usability Engineering**. Boston: AP Professional, 1993 362 p.
- [NZL 00] **New Zealand Digital Library**. Biblioteca Digital da Universidade de Waikato. Nova Zelândia. Material online. Online. Capturado em jul. 2000. Disponível na Internet: <http://www.nzdl.org/>
- [SHN 97] SHNEIDERMAN, Ben. **Designing the user interface: strategies for effective human-computer interaction**. 3. ed. Reading, MA: Addison-Wesley, 1992.
- [STR 96] STRZALKOWSKI, T., PEREZ-CARBALLO, J., MARINESCU, M. Natural Language Information Retrieval in Digital Libraries. In: **PROCEEDINGS OF ACM INTERNATIONAL CONFERENCE ON DIGITAL LIBRARIES**. 1. **Proceedings...**New York: ACM Press, 1996. p. 117-125.
- [WEB 00] **WEB ARTEMIS**. Recommendation System Guide. University of Michigan Digital Library. Capturado em ago. de 2000. Online. Disponível na Internet <http://agent.umd.umich.edu/artemis/guide/index.html>.
- [WIN 99] WINCKLER, Marco Antônio. **Proposta de uma metodologia para avaliação de usabilidade de interfaces WWW**. Porto Alegre, 1999. 97 f. Diss. (Mestrado em Ciência da Computação) - UFRGS, Instituto de Informática.

Proposta e validação de método para estudo da percepção de signos por sujeitos inseridos em diferentes meios culturais

Adéle Malta Pontes, Afonso Inácio Orth

Universidade Católica de Pelotas Pontifícia Universidade Católica do RS

Deli@atlas.ucpel.tche.br Orth@inf.purs.br

Resumo. O trabalho parte da idéia de que o modelo de percepção dos seres humanos é influenciado pelo contexto cultural no qual estão inseridos. Partindo deste princípio é proposto um método que visa possibilitar conhecer este modelo de percepção para que se espelhe nos signos de interface as características levantadas. O método foi validado em um estudo de casos entre sujeitos Surdos e Ouvintes.

1. Introdução

A crescente popularidade do computador e sua integração nas mais variadas atividades humanas tem aumentado a exigência de reflexões mais qualificadas em relação aos signos, representados nos ícones das interfaces de software.

O computador aparece como um instrumento que se utiliza de signos quando tenta prover o máximo de interatividade com seus usuários. Estes signos usados na interface dos programas relacionam objetos e tarefas da vida cotidiana dos usuários com as funções do sistema computacional.

Esta utilização dos signos nas interfaces vem aumentando a possibilidade de uso do computador como meio, no qual o usuário final não precisa ser um especialista, pelo contrário, o usuário, com seus conhecimentos simples, pode conseguir a interação e gastar seu tempo e esforço mental na atividade produtiva, típica do seu meio de trabalho e não na compreensão do sistema. Mas, para que a comunicação se estabeleça da forma interativa esperada é preciso que o projetista, a partir do estudo e do conhecimento das atividades que o programa pretende realizar, do contexto social de que faz parte e a que usuários se destina, forme um modelo comum a todos estes aspectos e represente-o por meio de signos apropriados.

Então, a idéia trabalhada, neste artigo, possui sua origem na suposição de que o modelo de percepção dos seres humanos é influenciado pelo contexto cultural em que a pessoa está inserida. É importante enfatizar que apesar de citar alguns autores, o trabalho não tenta realizar uma abordagem semiótica, mas considerar bibliografias de diversas áreas como é o caso da comunicação visual [BOR91], [POR97], a qual introduz o papel dos signos na comunicação humana, da área de psicologia Cognitiva e Social [KRE76]; [ROT98] que, além de atribuir ao meio social o modelo de percepção das pessoas, afirma que o contexto lingüístico também é um fator de grande importância na maneira como as pessoas percebem os objetos e, ainda, da área de Interação Homem-Computador [FRA89], [EVE97], [MAN97], que apresenta algumas pesquisas que mostram existir diferença cultural na maneira como as pessoas interagem com os computadores.

2. Signos X Modelo cultural

Segundo Bordenave [BOR91], os signos são qualquer coisa ou estímulo físico utilizados para representar objetos, qualidades, idéias ou eventos. Os signos, então, são o meio pelo qual as mensagens fluem de uma fonte para um destino.

A tentativa de projetar signos com características significativas para os usuários, parte do pressuposto de que a interface com o usuário é um artefato cultural [MAR99] e que os signos disponíveis nela tentam expressar as capacidades ou características de um objeto ou idéia, mas para isto se valem dos códigos e símbolos de utilização coletiva da sociedade em que está inserido, quer dizer, os signos devem ser significantes à cultura do grupo a que pretendem transportar qualquer tipo de informação.

Esta idéia pode ser evidenciada no texto de Bordenave, que diz que os signos não tem valor único e geral. Eles deflagram com maior ou menor probabilidade um determinado significado de acordo à relevância – para a vida do povo – dos objetos que eles representam na cultura circundante.

Para um melhor entendimento, ao termo “cultura”, é possível fazer referências ao conjunto de relações que o homem entretém com o meio natural e social em que vive, de acordo com sua percepção.

3. Problemática: Design de ícones de interface que reflitam características culturais

Pesquisas de interfaces usuário-computador como mediadora entre pessoas e máquinas não têm explorado verdadeiramente diferenças culturais entre usuários como sendo um fator crítico na determinação de como um software pode ser melhor acomodado aos diferentes grupos e necessidades de usuários.

Isto significa que em contrapartida ao projeto de sistemas que visam o uso internacional, o projeto de interfaces de softwares que visam sua utilização em um determinado mercado ou visam sua utilização por um determinado tipo de usuário, deve respeitar convenções locais e diferenças culturais na representação gráfica dos signos, quer dizer, no momento do projeto ou da seleção de ícones existentes, a cultura da população de usuários do produto deve ser considerada, caso seja desejado ao projeto um signo que represente adequadamente um objeto ou função. Por exemplo, enviar e receber e-mail é uma atividade internacional, mas representar esta funcionalidade por meio de um ícone de “caixa de correspondência” pode ser inapropriado, pois a aparência e forma destas caixas variam de país para país. Já representar esta atividade com um envelope pode ser uma boa solução, eis que este objeto é familiar a todas as culturas [GAL90] .

Além dos problemas com a intuitividade em relação aos ícones, pode acontecer “agressões” a uma determinada cultura, isto pode significar a completa rejeição do sistema. Temos como exemplo: O uso de um copo de *cocktail* em um sistema de agenda para representar um encontro marcado após o trabalho. Este ícone não deve ser apropriado em países onde o álcool não é associado com atividades sociais. Outro exemplo de um ícone inadequado, que tem duas diferenças de significado dependendo da cultura, é o uso de um gato preto como um símbolo para sorte. Nos estados Unidos um gato preto é associado com má sorte, mas no Reino Unido, um gato preto é considerado, justamente o contrário, ou seja, boa sorte.

A importância de levar em consideração a cultura dos usuários também é enfatizada por Portugal quando afirma que “A adequação cultural é crítica para o sucesso de um produto. Este deve ser o ponto de partida e não o ponto final do desenvolvimento, e deve ser o alicerce para a adequação funcional, ergonômica e cognitiva.” [POR97]. Esta adequação é uma forma poderosa de alcançar a atitude de satisfação no usuário em relação ao uso de sistemas, bem como a maneira efetiva de detectar as preferências de *design*, as crenças sobre a utilidade do sistema e a percepção do usuário sobre facilidade de uso.

4. Método para levantamento de características culturalmente significativas

O quadro referencial, adotado neste trabalho, parte do reconhecimento de que a cultura e o mundo em que vivemos são condicionantes na maneira como construímos a realidade

Nesta perspectiva, o método aqui proposto se concentra na descoberta de expectativas ocultas no pensamento dos usuários e nos seus meios culturais. Mediante as atividades a seguir apresentadas, acredita-se que é possível obter informações sobre preferências que o usuário não sabe informar de maneira direta. O trabalho também procura por categorias e princípios de organização para incorporá-las aos projetos.

O método pode ser representado a partir da seguinte seqüência de atividades[PON00]: a) construção e aplicação de um instrumento de reconhecimento visual; b) Análise qualitativa das respostas ao instrumento; c) Identificação de características aplicáveis; d) Análise quantitativa.

As atividades propostas no método são uma combinação de técnicas já utilizadas e validadas em outras áreas de conhecimento e que aqui são adaptadas para sua efetiva contribuição ao contexto computacional. Estas atividades serão, resumidamente, apresentadas a seguir.

1ª Fase – Construção e Aplicação de experimentos visuais

Demonstrações perceptuais representam o mais tradicional método de estudo da percepção visual. Ele consiste na apresentação, para determinados sujeitos, de figuras simples e perguntas diretas sobre o que eles vêem. Este método requer que sujeitos relatem suas introspecções e procura, nestes relatos, informações importantes sobre o que abrange a natureza da tarefa do sistema visual.

Contudo, segundo Roth [ROT98] demonstrações perceptuais apresentam dois problemas como método para investigação da percepção visual. Primeiro, enquanto os sujeitos percebem a natureza do fenômeno a ser interpretado, acabam por não interpretar o fenômeno, propriamente dito, segundo, introspecções são potencialmente inexatas e, muitas vezes, não demonstram o que realmente o sujeito percebeu.

Como um caminho de solucionar esses problemas aparecem os experimentos. Eles, por sua vez, se referem às demonstrações combinadas à possibilidade de realizar o estudo das respostas dos sujeitos. Este estudo visa selecionar, cuidadosamente, tipos de informações visuais, apresentadas sob condições cautelosamente controladas. Neste método, várias técnicas são usadas para mensurar as respostas dos sujeitos e as conclusões são, usualmente, baseadas em uma média de respostas de um grupo, mais do que em respostas individuais. A maioria dos experimentos contemporâneos se ocupam de

estímulos visuais, baseados em modelos encontrados no mundo real, em que sujeitos são instigados a responder a tais estímulos. Por exemplo, nomeie isto; categorize aquilo, relacione isto com aquilo.

Esta primeira etapa, então, propõe a construção e aplicação de um instrumento de experimentação visual, considerando o contexto e escopo que se pretende levantar informações.

2ª Fase - Análise qualitativa dos dados obtidos com a aplicação do instrumento

Após a aplicação do experimento visual, sugere-se a análise das respostas dos sujeitos. A técnica de análise proposta denomina-se Análise de conteúdo e representa uma exploração qualitativa de mensagens e informações, permitindo atingir uma compreensão aprofundada dos significados das mensagens, bem como *“permitindo o conhecimento de fenômenos da vida social de outro modo inacessíveis”* [OLB89].

Isso quer dizer que a Análise de conteúdo, além de permitir a interpretação do conteúdo explícito das mensagens, permite a análise do conteúdo oculto, quer dizer, oferece meios de interpretar fatos e idéias presentes nas mensagens, mas que, de outra forma, não seriam descobertas ou explicitadas. Também pareceu relevante, na escolha desta técnica, o fato do processo tratar o produtor das mensagens (neste caso os sujeitos entrevistados) como um produto social, condicionado pelos interesses da classe a que pertence e que, por isto, evidencia em suas mensagens, sejam elas faladas, escritas ou sensoriais, suas concepções de mundo, interesses de classe, traços psicológicos, motivações, expectativas e outras, e que em suas mensagens sempre estão descritos, mesmo que de forma inconsciente, o que consideram mais importante sobre sua concepção da realidade.

Forma de aplicação da Análise de conteúdo, conforme proposto em Moraes [MOR99]:

Preparação das informações: significa identificar dentre as amostras a serem analisadas, quais efetivamente, são representativas em relação aos objetivos da pesquisa.

Transformação do conteúdo em unidades de Análise: definir e codificar “unidades de significado”, que podem ser palavras, frases ou temas.

Categorização: este é um procedimento de agrupar dados, considerando a parte comum existente entre eles. Classifica-se por semelhança ou analogia, segundo critérios previamente definidos.

Descrição: é comunicado o resultado da categorização, por meio de um texto síntese de cada uma das categorias. É o momento de expressar os significados, captados e lidos nas mensagens analisadas.

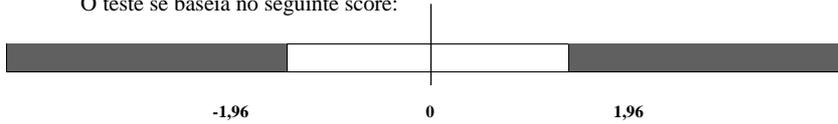
Interpretação: Nesta fase, é importante que se procure atingir uma compreensão mais profunda do conteúdo das mensagens mediante a inferência e a interpretação, mais aprofundada, não só sobre conteúdos manifestos como, também, sobre os latentes, sejam eles ocultos, consciente ou inconscientemente, pelos autores.

3ª Fase – Análise quantitativa dos dados obtidos na fase anterior

Como forma de fazer descrições mais sucintas e obter inferências a partir dos dados, de maneira mais sistemática e segura, recomenda-se a utilização do teste de diferença de proporções. Este é um método estatístico e apresenta sua verdadeira importância como um instrumento auxiliar na investigação de características sem limitar ou mutilar a análise feita anteriormente.

O método é utilizado para demonstrar se existe significativa diferença entre as escolhas feitas pelos usuários analisados, ou seja, por meio da quantificação pretende-se alcançar respostas para questões como: As características que emergiram e que deram origem às categorias são aceitas e valorizadas de forma similar ou de forma diferenciada entre os grupos? Quando comprovada a diferença, ela é realmente significativa a ponto de se poder distinguir características relevantes a um ou outro grupo? E em alguma característica, existe convergência entre os grupos?

O teste se baseia no seguinte score:



Há diferença de proporções

Não há diferença de proporções

4. Validação do método proposto

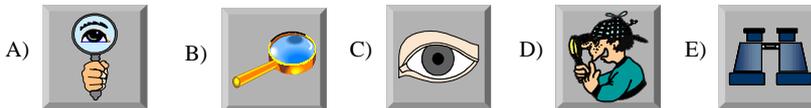
Este método foi validado em um estudo de caso a partir de 189 depoimentos escritos coletados junto a dois grupos de alunos de 6ª série, sendo um de alunos surdos e outro, de alunos ouvintes, eis que estes grupos de usuários podem ser considerados inseridos em grupos culturalmente distintos [PON00].

4.1 Experimento Visual

O desenvolvimento do instrumento de experimentação visual para aplicação no estudo de caso, se baseou em alguns atributos, relacionados à construção de ícones, indicados por Mandel [MAN97]: a) Significância do ícone; b) Identificação do ícone; Eficácia do ícone. Neste caso particular o instrumento foi desenvolvido em três partes, onde cada uma delas se baseou em um destes atributos.

Exemplo da estrutura da primeira parte do experimento de percepção visual utilizado no estudo de caso.

- a) Um conjunto de ícones, indicados por letras do alfabeto romano, escolhidos e editados para apresentar algum tipo de relação com a metáfora “procurar”;



- b) A metáfora chamada, no instrumento, de palavra;

Palavra: **Procurar**

- c) Instruções para que se fizesse a escolha de um dos ícones que melhor combinasse/representasse a palavra dada e para que se justificasse esta escolha.

1º) *Observe com atenção os seguintes desenhos:*

2º) *Indique qual deles combina mais com a palavra descrita abaixo.*

Opção:()

3º) *Explique **porque** escolheu este desenho, descrevendo o que lhe chamou atenção para que ele fosse escolhido entre os demais:*

4.2 Análise de Conteúdo

Em cada um dos depoimentos foi realizada a análise de conteúdo. A aplicação desta análise seguiu aos procedimentos propostos por Moraes [MOR99], já citados acima. É importante ressaltar que as escolhas propriamente ditas não são analisadas em profundidade, mas sim as justificativas pela escolha. É o “Porquê” dado pelo sujeito que interessa na análise qualitativa.

Da análise de Conteúdo aplicada aos depoimentos emergiram 5 categorias principais, reunindo idéias que expressam o que levaram os sujeitos a fazer sua escolha no instrumento de percepção

Tabela 1 – Categorias Definidas

Categorias	
1	Relação com a representação de movimento
2	Relação com a presença da figura humana
3	Relação com a completeza (movimento + figura humana + instrumento)
4	Relação com a linguagem vocabulário
5	Relação com instrumento/experiência

Fonte: [PON00]

Algumas considerações merecem serem feitas em relação a estas categorias:

Relação com a representação de movimento

Uma concepção que emerge com grande intensidade da análise das respostas, no grupo dos Surdos, é a que concebe a escolha do ícone como uma forma de representação

da manifestação de uma força ou ação. Isto significa que os Surdos preferem símbolos que representam uma certa dinâmica. Neste estudo foi possível perceber que o que parece importar para o grupo dos Surdos, é a aparência de animação na figura, mesmo que ela esteja representada de forma indireta. Muitos ícones que não foram projetados, intencionalmente, para representar movimento, foram percebidos por este grupo, como figuras móveis. Sem a preocupação de explicar o porquê ou como determinado ato ocorre, muitos depoimentos demonstram relação com uma ação que parece estar acontecendo naquele determinado instante. Para ilustrar esta afirmação pode-se destacar as expressões: “*está pintando*” , “*Algo está sendo jogado fora.*””, ou ainda “*procurando muito*”.

Relação com a presença da figura humana

Um número significativo de depoimentos, tanto de Surdos, quanto de Ouvintes, aborda a questão da presença da figura humana nos ícones apresentados. Figura humana aqui envolve a aparição de um indivíduo ou partes que lembram membros de um corpo humano (braço, mão, olho, etc.), os quais fizeram os sujeitos se identificarem nas figuras. Esta perspectiva mostra-nos a importância de deixar o sujeito ser “aquele que determina um acontecimento”, ou seja, agrada ao usuário a possibilidade de acordo com sua vontade, controlar ou não, o acontecimento de uma determinada ação. Esta categoria foi fundamentada na aparição de pronomes como: “eu”, “ele”, “aquele”, em nomes próprios como: “Paulo” e em substantivos como: “homem” e “trabalhador”.

Relação com a completeza (movimento + figura humana + instrumento)

Manifestada por um número significativo de sujeitos, a relação com a completeza do desenho está inteiramente relacionada com a característica de “que não falta nada do que pode ou deve ter um ícone”. Quer dizer, a aparição da sensação de movimento, mais a manifestação da figura humana, e a representação de um instrumento se complementam, formando um sentimento de satisfação, melhor dizendo de aceitação do complexo, que concerne na qualidade sintática [SAR98] que diz respeito às combinações dos atributos visuais, tais como aparência e movimento (linhas, padrões, tamanho e formato) que determinam a facilidade com que signos visuais podem ser distinguidos e identificados.

Exemplo: “Ele está excluindo o lixo na lata.”

Relação com a linguagem/vocabulário

Esta categoria representa a importância da escolha de palavras e termos baseada no conjunto de palavras e significados existentes na língua natural dos usuários. Ficou evidente, nesta categoria, que a habilidade em relacionar um ícone com um significado está, inteiramente, relacionado à habilidade em identificar este significado em um contexto lingüístico conhecido. Quer dizer, que no processo de relacionamento da figura com o significado, há a intervenção direta do vocabulário existente na linguagem utilizada pelo sujeito Surdo. Então, todas as representações de signos na interface devem visar a adequação das representações à luz dos conhecimentos lingüísticos dos usuários. Esta característica foi marcante no grupo e Surdos, pois não tinham conhecimento de palavras como: “excluir”.

Relação com instrumento/experiência

A palavra instrumento, nesta subcategoria, é abordada como qualquer objeto empregado na execução de alguma atividade.

A palavra experiência foi adicionada para representar a relação de uso destes instrumentos, em práticas contínuas de uma arte ou ofício. Isto significa que as escolhas realizadas que se enquadram nesta categoria estão vinculadas à aparição de certos instrumentos que, de alguma forma, já haviam sido utilizados pelos sujeitos, ou já tinham tido sua utilização presenciada. Esta afirmação pode ser evidenciada pelas expressões: “por que é o jeito de pintar”, “Por que se pinta com lápis de cor” , “Por que só com pincéis eu consigo pintar” e “Está muito certo usar saco de lixo lata.”.

Essas categorias são provenientes de uma análise das justificativas escritas pelos respondentes e, por emergirem destes depoimentos, mostram-se como um caminho para ajudar na produção de signos, uma vez que, segundo Saraiva, “ *o mecanismo da invenção não é suficientemente claro*” [SAR98], este tipo de categorização vem também em acordo a idéia de Eco apud [SAR98] que diz que o mais razoável é analisar o mecanismo pelo qual as metáforas são interpretadas e, a partir dele, procurar conjecturar as possíveis faces de sua geração.

4.3 Análise quantitativa

A Tabela 2 mostra o resultado do Teste da diferença de proporções, propriamente, dito. Onde:

FS - Freqüência de aparição de depoimentos de Surdos na categoria;

FO – Freqüência de aparição de depoimentos de ouvintes na categoria.

Zc – Proporções propriamente ditas.

Tabela 2 – Resultado do Teste da diferença de proporções

CATEGORIAS	FS(%)	FO(%)	Zc	Resultado
Relação com a representação de movimento	31,74	18,26	2,0800	Há diferença significativa
Relação com a presença da figura humana	7,93	11,11	0,9366	Não há diferença significativa
Relação com a completeza (movimento + figura humana + instrumento)	25,40	26,20	-0,1194	Não há diferença
Relação com a linguagem/vocabulário	4,77	0	8,5791	Há diferença significativa
Relação com instrumento/experiência	19,05	33,33	-2,0635	Há diferença significativa

Fonte: [PON00]

Realizando algumas considerações finais referentes ao Teste de diferença de proporções, é possível concluir que há diferença significativa na valorização de algumas características, apresentadas nos signos, por Surdos e Ouvintes. Surdos valorizam, em primeiro lugar, a característica de representação do movimento, enquanto ouvintes valorizam, preferencialmente, a representação de um instrumento, relacionado à sua

experiência anterior. Os dois grupos valorizam, da mesma forma, a questão da presença da figura humana e convergem, de forma significativa, na valorização de signos que apresentem a característica da completeza.

Acredita-se que, ao considerar as características representadas nestas categorias, estar-se-á dando ao usuário Surdo a nítida sensação de que a interface foi realmente projetada para ele. E, de acordo com Orth [ORT93], o usuário sentindo-se confortável no uso da interface, é provável que até aceite algumas inadequações ergonômicas ou funcionais ou se disponha a fazer um esforço adicional para conseguir memorizar ou aprender a usar algum comando ou executar algum procedimento menos ajustado as suas características.

5 Exemplo da aplicação das características levantadas

Os signos apresentados a seguir são exemplos de como incorporar as características valorizadas pelos usuários, sejam eles Surdos ou Ouvintes, eis que foi possível encontrar uma categoria comum aos dois grupos de usuários. Note-se que não houve preocupação em mudar as metáforas já utilizadas e praticamente consagradas como padrões de interface, a preocupação maior foi em adicionar as características de movimento e/ou representação da figura humana aos ícones projetados.



6. Considerações Finais

As características percebidas a serem implementadas nos signos mostram-se como uma forma de não causar dissonâncias culturais, além de uma forma de tornar os signos mais atrativos e intuitivos aos usuários.

Para alcançar este objetivo o trabalho apresentou a proposta da integração de técnicas provenientes de outras áreas, em um método para o estudo dos usuários inseridos em diferentes meios culturais e suas preferências de *Design*, eis que, isto sempre se impõe como uma barreira no momento do projeto dos signos das interfaces de software.

O método ao invés de submeter os usuários a testes e observações de protótipo, utiliza um experimento de reconhecimento visual, combinado com os paradigmas qualitativo e quantitativo de análise. A combinação das análises foi escolhida, pois a análise qualitativa permite perceber idéias implícitas nas escolhas e relatos dos entrevistados, característica que, por meio de uma análise quantitativa, não seria possível

conservar. A análise quantitativa, por sua vez, permite o teste da hipótese levantada de forma objetiva.

O método teve sua validação em um estudo de caso com sujeitos Surdos e Ouvintes, da seguinte forma: aplicação do experimento visual, Análise de conteúdo, teste de diferença de proporções e o *design* propriamente dito de ícones de interface [PON00].

Referências

- [BOR91] Bordenave, J. E. D.; Além dos meios e mensagens, Introdução à comunicação como processo, tecnologia, sistema e ciência. 5 ed.. Petrópolis: Vozes, 1991.
- [EVE97] Evers, V; Day, D. - The role of Culture in interface Acceptance. Human-Computer Interaction: Interact'97. Chapman & Hall, 1997.
- [FRA89] Franciosi, B. R. T; Cláudio, D. M.; Projeto de Interfaces gráficas. Congresso nacional de informática, XXII, 1989. Anais. São Paulo, 1989.
- [GAL90] Galdo, E. D.; Internationalization and translation: some guidelines for the design of Human-computer Interfaces. Designing user Interfaces for International Use, ed. by J. Nielsen Elsevier, 1990.
- [KRE76] Krech, D.; Crutchfield, R. S.; Elementos de psicologia. 5ª edição, São Paulo: Livraria Pioneira, 1976.
- [MAN97] Mandel, T.; Elements of User Interface Design. John Willey & Sons, 1997.
- [MAR99] Marcus, A.; Improving the User Interface. Disponível em: <http://www.webword.com/interviews/marcus.html>
- [MOR99] Moraes, R.; Análise de Conteúdo. Educação, porto Alegre, ano XXII n° 37, março 1999. P. 7-32.
- [Olb89] Olbuenaga, J. I. R.; Ispizua, M.; La descodification de la vida cotidiana: metodos de investigacion cualitativa. Bilbao: Universidad de Deusto, 1989.
- [ORT93] Orth, A. I.; Interfaces Homem-máquina: A Personalização do Atendimento ao usuário .Tese de Doutorado, COPPE/UFRRJ, 1993.
- [PON00] Pontes, A.; Estudo da percepção de signos por sujeitos inseridos em diferentes meios culturais: Um passo inicial para o desenvolvimento de uma interface voltada a usuários Surdos. Dissertação de Mestrado, Faculdade de Informática, PUCRS, 2000.
- [POR97] Portugal, S.; Visual Interaction design: design as a cultural activity. SIGCHI Bulletin vol 29 n° 3 July 1997.
- [Rot98] Roth, I.; Bruce, V.; Perception and Representation, Mcgraw-Hill, inc 1998.
- [sar98] Saraiva, N. A.; Semiótica e comunicação: Do texto à interface do programa. Dissertação de mestrado em semiótica. Curso e pós-graduação em ciências da comunicação. UNISINOS. São Leopoldo, julho, 1998.

Reengenharia de Interfaces com Realidade Virtual Considerando o Paradigma de Orientação a Objetos

Cristiano L. Soares, Júnia C. Anacleto Silva, Rosângela D. Penteadó

Universidade Federal de São Carlos - Departamento de Computação
São Carlos, SP, Brasil, 13565-905

anechini@yahoo.com, {junia,rosangel}@dc.ufscar.br

Abstract. *The proposition of this paper is to evaluate the Objects-Oriented paradigm (OO) in order to support the interfaces reengineering process along with the support of the GaCIV environment. The sample system used is considered in three versions: legacy code (originally implemented using Clipper), segmented code (implemented using Clipper with OO characteristics) and that one which went through the OO reengineering process (implementation using Java language). Key-words: OOUI, VR and interfaces reengineering*

Resumo. *Este artigo tem como proposta verificar a adequação do paradigma orientado a objetos (OO) para apoiar o processo de reengenharia de interfaces juntamente com o ambiente GaCIV. O sistema exemplo utilizado é considerado em três versões: legado (originalmente implementado em Clipper), segmentado (implementado em Clipper com características OO) e que passou pelo processo de reengenharia OO (implementação em linguagem Java). Palavras-chaves: OOUI, RV e reengenharia de interfaces.*

1. Introdução

Com a evolução constante das tecnologias aplicadas a interfaces do usuário, uma nova geração de interfaces vem se estabelecendo. Dentre as novas tecnologias de interface surge com destaque a Realidade Virtual (RV) facilitando a visualização, manipulação e interação de usuários com ambientes computacionais tridimensionais (3D). Além da RV, a aplicação do paradigma de orientação a objetos (OO) apresenta-se como uma opção para desenvolvimento de interfaces não só intuitivas e naturais, mas também com possibilidade de reuso de seus componentes facilitando futuros desenvolvimentos.

Devido aos avanços na tecnologia de interface, muitos sistemas, apesar de ainda estarem funcionando de acordo com as necessidades de seus usuários, podem ter sua *performance* prejudicada em consequência de sua interface ultrapassada. Para solucionar problemas desse tipo, a melhor solução é a prática da reengenharia da interface do sistema com o apoio de ferramentas automatizadas quando possível. A reengenharia de interfaces pode tornar os sistemas legados mais amigáveis e dessa forma aumentar o seu tempo de vida.

Entretanto, o uso de interfaces avançadas, seja em um processo de engenharia avante ou em um processo de reengenharia de interfaces, ainda adiciona complexidade ao desenvolvimento de sistemas interativos devido à falta de ferramentas, métodos e diretivas de apoio.

Este trabalho tem, entre outros objetivos, apresentar diretrizes para a reengenharia de interfaces OO com recursos de RV utilizando o ambiente GaCIV (Gabaritos Configuráveis para elaboração de Interfaces com RV) [22] como apoio.

Este artigo está organizado da seguinte maneira: a seção 2 apresenta uma visão geral do desenvolvimento de interfaces; uma breve análise de metodologias para o desenvolvimento de interfaces é mostrada na seção 3; na seção 4 será apresentado o ambiente GaCIV como ferramenta de apoio para desenvolvimento de interfaces com recursos de RV; a seção 5 apresenta a integração entre engenharia reversa e reengenharia com recursos de RV; o estudo de caso utilizado neste trabalho é mostrado na seção 6; na seção 7 são apresentadas as considerações finais; na seção 8 são apresentados os agradecimentos e, finalmente, na seção 9 a bibliografia.

2. Visão Geral do Desenvolvimento de Interfaces

Durante toda a história da evolução das técnicas de interface, três estilos bem distintos caracterizaram três gerações de interfaces. Cada uma dessas gerações foi definida pela tecnologia de hardware existente na época [24].

Entre a década de 1950 e 1960, a primeira geração de “interfaces” era formada por cartões perfurados e computadores que só funcionavam em modo *batch*, sem haver interação real entre homem e máquina. A segunda geração de interfaces, durante o início da década de 1960, foi marcada pelo uso de mainframes e minicomputadores, através da técnica de *timesharing* com monitores monocromáticos e interfaces textuais. Usuários podiam então interagir com o computador digitando comandos com parâmetros.

A geração seguinte surgiu das pesquisas da Xerox PARC durante a década de 1970 [24]. Estações de trabalho ligadas em rede com monitores gráficos e WIMP GUI's (*Windows, Icons, Menus, Pointin Device – Graphical User Interface*), que segundo [24] são interfaces gráficas baseadas em janelas, ícones, menus, e um dispositivo de seleção, tipicamente um *mouse*.

A quarta geração de interfaces, segundo [24], é formada por interfaces que implementam pelo menos uma interação independente de *widgets* WIMP. Essas interfaces são chamadas de Pós-WIMP e têm por objetivo: envolver diversos sentidos humanos em paralelo; tornar a interação mais natural e intuitiva; permitir a interação de múltiplos usuários simultaneamente; e aumentar a liberdade em relação à proximidade física do usuário e sistema.

A tecnologia de interação de quarta geração mais difundida é a RV. Tem como principal característica exploração de diversos sentidos do usuário para causar uma sensação de imersão do usuário em um ambiente virtual onde o usuário manipula objetos e realiza tarefas semelhante ao que é feito no mundo real [26]. Dessa forma, a realidade virtual torna a interação do usuário mais intuitiva e natural.

As mudanças propostas pela RV e outras tecnologias de interface Pós-WIMP causam um impacto direto à forma de desenvolvimento de interfaces. A implementação de tais interfaces ainda apresentam alto grau de complexidade [7] e custo em relação às interfaces WIMP.

Desde a constatação do importante papel representado pela interface em sistemas interativos durante a Segunda Guerra Mundial, diversas teorias, modelos e métodos foram criados para o estudo e desenvolvimento de interfaces.

Dentre esses destaca-se o Modelo Cognitivo que apresenta uma base teórica, fundamentada na psicologia cognitiva, para o entendimento de como o usuário se comporta diante de uma interface de computador. A partir desse modelo, vários métodos de desenvolvimento, teste e avaliação de interfaces foram criados tais como o CLG (*Command Language Grammar*) [16], o TAL (*Task Action Language*) [19] e o GOMS (*Goals, Operators, Methods and Selection rules*) [4]. O GOMS faz uso de um modelo do comportamento humano, o MPH (Modelo do Processador Humano), formulado a partir de metáforas computacionais para representar teorias psicológicas.

Além da abordagem cognitiva, foram desenvolvidas e/ou adaptadas abordagens computacionais para o desenvolvimento de interfaces e sistemas interativos. Com isso surgiu a abordagem do Projeto Centrado no Usuário, em que o desenvolvimento dá prioridade às necessidades, às limitações, enfim, às características dos usuários do sistema [10], como é o caso das metodologias USE (*User Software Engineering*) [25], LUCID (*Logical User Centered Interaction Design*) [5] e a Engenharia de Usabilidade [17]. Outra abordagem de desenvolvimento de sistemas interativos é o Projeto Participatório [14][10], no qual as preocupações em relação à satisfação do usuário demonstradas na abordagem Centrada no Usuário persistem, porém, com a característica adicional da participação ativa do usuário no desenvolvimento do sistema em uma ou mais fases do processo [10]. Muitas metodologias foram desenvolvidas seguindo essa abordagem, como a metodologia *The Bridge* [8] na qual se percebe as características básicas de Projeto Participatório [14].

Muitos conceitos surgiram a partir da década de 40 e amadureceram dando origem ao termo “usabilidade” da interface. Usabilidade define um conjunto de características desejáveis para que a interface possa prover acesso às funções de uma aplicação de modo que reflita a maneira de pensar do usuário [27]. Isso requer que a aplicação não só forneça suporte para aspectos necessários para o trabalho do usuário, mas também deve fornecer meios para interagir com o sistema de forma intuitiva e natural [27]. Grande parte desses conceitos está incluída no Projeto Centrado no Usuário e no Projeto Participatório.

Atualmente, a IBM utiliza o termo “*ease-of-use*” ou “facilidade de uso” com o mesmo significado de “usabilidade” e busca desenvolver metodologias para o desenvolvimento de interfaces que tenham garantida essa característica [12].

Mais recentemente, no final da década de 1980, o grupo de IHC (Interação Humano-Computador) da IBM percebeu que os usuários não se beneficiavam com os sistemas operacionais caracterizados pela capacidade de abrir diversas janelas e permitir que vários aplicativos executassem “ao mesmo tempo”. Pelo contrário, os usuários ficavam sobrecarregados e não obtinham nenhuma ajuda do computador para gerenciar os aplicativos [12]. Essa percepção levou ao desenvolvimento das OOUI's (interfaces Orientadas a Objetos). Interfaces OO é o termo utilizado para designar interfaces nas quais o usuário reconhece representações de objetos reais relevantes à sua tarefa, permitindo organizá-las e agir diretamente sobre essas, na tentativa de realizar suas tarefas de forma similar à realizada no mundo real [6].

3. O Ambiente GaCIV Como Apoio ao Desenvolvimento de Interfaces Pós-Wimp

As mudanças propostas pela RV e outras tecnologias de interface Pós-WIMP causam impacto direto na forma de desenvolvimento de interfaces. A implementação de tais interfaces ainda apresenta alto grau de complexidade e custo em relação às interfaces WIMP. O ambiente GaCIV (Gabaritos Configuráveis para elaboração de Interfaces com RV) [1] [22], ferramenta em desenvolvimento no Departamento de Computação da Universidade Federal de São Carlos, tem como objetivo dar apoio à criação de interfaces Pós-WIMP de forma rápida e viável.

Os principais objetivos desse ambiente são:

- facilitar o uso da tecnologia de RV não-imersiva para a criação de interfaces dessa natureza, permitindo a criação de interfaces através da manipulação direta de objetos e ambientes para configuração de uma interface;
- facilitar a aproximação entre projetistas de interfaces, engenheiros de software e usuário durante o desenvolvimento de uma aplicação com interfaces RV aproveitando, para isso, a própria natureza intuitiva da visualização de ambientes RV, de forma que permita melhor comunicação entre os participantes da equipe de desenvolvimento da aplicação.

As interfaces RV construídas com a utilização do GaCIV são geradas por gabaritos configuráveis. Segundo [1], um gabarito é um ambiente virtual que contém um conjunto de objetos que o caracterizam. Os objetos que fazem parte do ambiente virtual podem ser escolhidos e configurados de acordo com as especificações do usuário. É possível também criar uma biblioteca de gabaritos para ser usada pelo projetista de interfaces, que terá a oportunidade de escolher a interface mais adequada para uma determinada aplicação e também escolher a melhor configuração dessa interface de acordo com as necessidades do usuário.

Segundo [2], o GaCIV aplica o conceito de RV não-imersiva para oferecer os recursos de RV de forma acessível, superando restrições de equipamento. Para atingir os objetivos de suporte à criação de interfaces RV de forma prática e acessível, o GaCIV permite três operações: criação de interfaces RV a partir de gabaritos, a principal entre elas; geração de novos gabaritos; e inserção de novos ambientes e objetos a serem usados em gabaritos.

Para a criação de interfaces com o GaCIV, a equipe de desenvolvimento deve escolher um gabarito da biblioteca de gabaritos. Os objetos que caracterizam o gabarito serão então apresentados para que o projetista possa selecionar quais serão utilizados para criação da interface. Ao ser escolhido, o objeto é inserido automaticamente em uma posição sugerida pela configuração padrão do gabarito. Os objetos inseridos podem ser modificados em relação à sua posição e tamanho, além de poder ser duplicados ou excluídos, permitindo assim a personalização da interface de acordo com os requisitos do sistema. Com os objetos devidamente inseridos no gabarito, o projetista obtém a interface virtual para o sistema em desenvolvimento [2]. Essa interface deverá ser devidamente conectada à aplicação como será visto adiante.

A operação de gerar novos gabaritos compreende a associação de um ambiente, existente na biblioteca do GaCIV, ao novo gabarito e posteriormente a adição de objetos, também previamente existentes na biblioteca de objetos, no gabarito criado. Quando finalizada a

configuração do gabarito, esse ficará armazenado na base de dados do GaCIV para ser usado na geração de novas interfaces RV. Essa operação, de geração de gabaritos, é uma operação de apoio à geração de interfaces RV.

A figura 1 mostra a tela do GaCIV durante a operação de criação de interfaces. Nessa tela podem ser vistos, à esquerda, duas listas. A primeira lista apresenta os objetos disponíveis no gabarito escolhido. A segunda lista mostra os objetos incluídos na interface. Dois botões, “Incluir” e “Excluir”, estão presentes entre as listas, com a finalidade, respectivamente, de inserir e de excluir objetos na interface em desenvolvimento. Ao lado direito da tela estão o nome da interface em desenvolvimento, o nome do ambiente utilizado e a visão 3D do gabarito.

Na barra superior são encontrados menus *pop up* e botões. Os menus apresentam opções para manipulação de arquivos, gabaritos, aplicações, ambientes, objetos e interfaces. Os botões logo abaixo da barra superior fornecem atalhos para abrir um gabarito já existente, sair do GaCIV e salvar gabarito. Os botões maiores (“Escala”, “Centraliza”, “Link”) oferecem atalhos para mudança de escala dos objetos, centralização de objetos e ligação de objetos a aplicações.

Novos objetos e ambientes também podem ser inseridos na base de dados do GaCIV para permitir uma maior gama de alternativas ao projetista de interfaces. Os objetos e ambientes a serem inseridos devem ter sido criados em editores 3D e, mais tarde, convertidos em arquivos do formato “nff”, que é o formato aceito pelo ambiente. Os nomes dos objetos e ambientes, após serem escolhidos e atribuídos pelos usuários, serão acrescentados na base de dados do GaCIV.

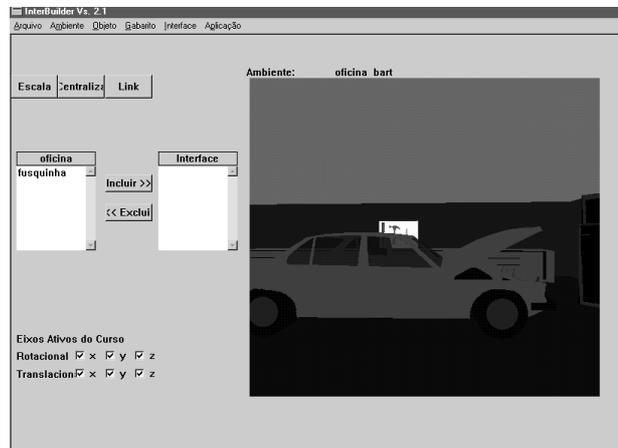


Figura 1 – Criação de interfaces no GaCIV

Como é mostrado em [2], o GaCIV integra os componentes de interface gerados pelo próprio GaCIV com aplicações desenvolvidas de forma independente, utilizando o recurso de *link*. O *link* é atribuído pelo ambiente a cada objeto da interface. Como visto em [1], um *link* é uma interligação entre o objeto da interface e um arquivo executável.

Dessa forma, quando um objeto da interface é ativado, uma aplicação associada a esse objeto é executada. Essa característica de associação de objetos a módulos da aplicação torna interessante a aplicação do paradigma de OO durante o desenvolvimento dos sistemas interativos, pois a OO facilita a modularização da aplicação [1] [23]. A figura 2 ilustra a ligação de um objeto com uma aplicação.

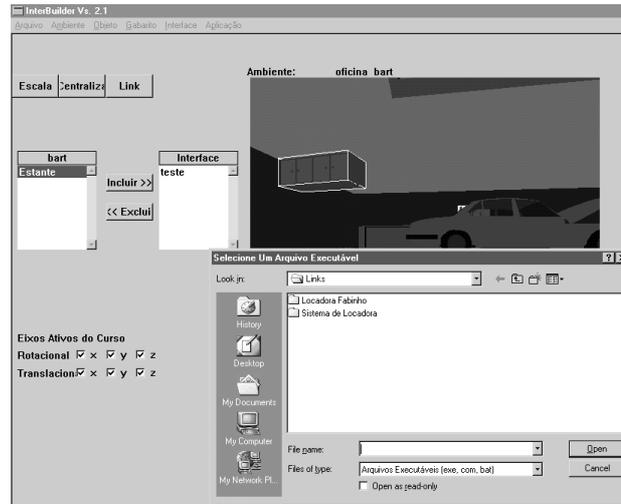


Figura 2 – Ligação entre objeto e aplicação no GaCIV.

A ação de ligação mostrada na figura 2 consiste em selecionar o objeto (no caso, o fogão), clicar no botão “Link” e escolher a aplicação que deve ser associada a ele. A janela central, vista na figura 2, mostra as diversas opções de aplicações existentes para o usuário ligar ao objeto fogão.

Segundo [2], com a capacidade de criar novos gabaritos e inserção de novos objetos e ambientes, o GaCIV permite a aplicação de um modelo de desenvolvimento de sistemas interativos baseada no Projeto Centrado no Usuário (UCD) [10].

O ambiente GaCIV pode ser utilizado também em processos de reengenharia de interfaces de sistemas legados (figura 3). O sistema legado será particionado de modo que a nova interface seja gerada com recursos de RV e usada para acessar as funcionalidades da aplicação, como mostra a parte da Figura 3 rotulada como “Nova Aplicação.exe”. Para que seja possível essa aplicação do GaCIV, é necessário que se faça uma análise do sistema legado buscando o entendimento das interfaces existentes e a separação de todo o código referente à interface desse sistema. Essa fase preparatória é necessária devido ao tratamento dado pelo GaCIV à independência de diálogo. Deve-se, portanto, preparar a aplicação para receber a nova interface criada no GaCIV. Ao mesmo tempo, deve-se realizar outras atividades de engenharia reversa buscando entendimento da aplicação para que a nova interface a ser desenvolvida seja equivalente à anterior em relação aos aspectos de funcionalidade. Devido à natureza modular da interface gerada pelo GaCIV [2], é desejável que a aplicação também possua essa característica para simplificar a associação da

interface criada no GaCIV à aplicação, permitindo identificar e isolar módulos da aplicação que serão disponibilizadas por objetos distintos da interface.

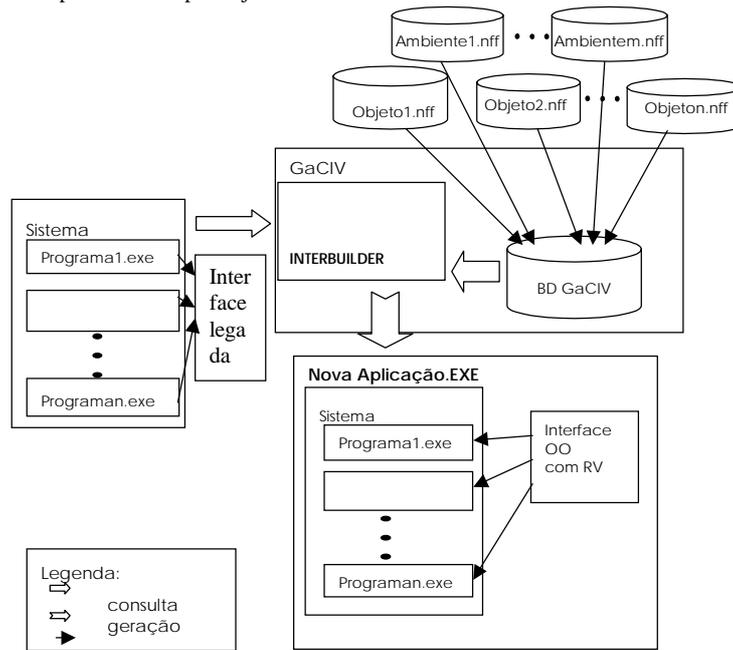


Figura 3 – O GaCIV e a Reengenharia de Interfaces.

4. Integrando Engenharia Reversa e Reengenharia com Realidade Virtual

Como visto anteriormente, projetos com interfaces avançadas são mais difíceis de se construir do que projetos com interfaces WIMP GUI's. A implementação de sistemas interativos seguindo o paradigma de OO é um aspecto importante [15], entre outros, pois acrescenta característica modular ao projeto. A utilização do paradigma orientado a objetos leva à menor carga de programação, principalmente devido a existência de classes, objetos e possibilidade de reutilização de código, inerentes a esse paradigma [12] [13]. Apesar disso, ainda há carência de métodos, técnicas e ferramentas computacionais de alto nível para apoio à criação de interfaces avançadas, como, por exemplo, interfaces com RV OO. Falta apoio para o processo de desenvolvimento de interfaces no qual haja cooperação entre projetista de interfaces, engenheiro de software e usuário para garantir a qualidade do projeto [21].

O objetivo deste estudo é verificar que a reengenharia de sistemas com a mudança de paradigma procedural para o paradigma da OO facilita o desenvolvimento de interfaces avançadas para tais sistemas, com apoio do ambiente GaCIV. Confirmar a adequação do paradigma de OO no processo de engenharia reversa e reengenharia de sistemas visando a implementação de interfaces avançadas com recursos de RV também faz parte da proposta deste trabalho.

A integração entre a engenharia reversa, reengenharia e a RV apoiada pelo GaCIV é realizada como na figura 4.

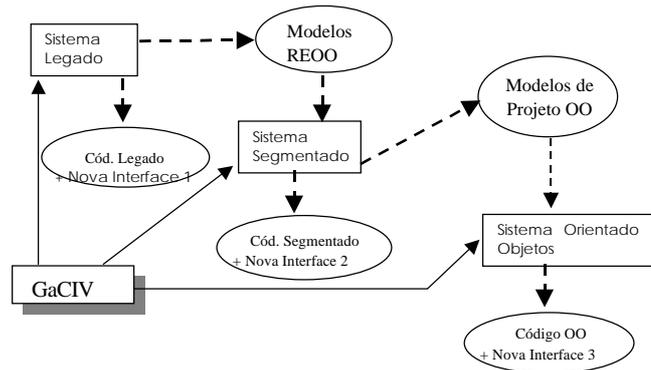


Figura 4 – Integração da atividade de engenharia reversa e reengenharia com a aplicação de recursos de RV.

Na figura 4 estão representados alguns sistemas e modelos já existentes e outros ainda a serem desenvolvidos. O sistema legado é um sistema comercial já existente que será comentado adiante. Os modelos de análise orientados a objetos (Modelos REOO) e a versão segmentada do sistema foram obtidos através de engenharia reversa e reengenharia anteriores ao início das atividades deste projeto [18]. A segmentação consiste no processo de reengenharia com mudança de paradigma de desenvolvimento, para o orientado a objetos, mas preservando a linguagem de implementação original. Os modelos obtidos na fase de engenharia reversa foram utilizados na segmentação e a partir desses, os modelos de projeto são construídos para a reengenharia com mudança de linguagem de programação para uma orientada a objetos, nesse caso, Java. Paralelamente, a reengenharia de interface das versões do sistema será realizada com apoio do ambiente GaCIV.

As versões legada, segmentada e a OO resultante da reengenharia serão consideradas para o desenvolvimento de novas interfaces através do ambiente GaCIV verificando-se a viabilidade do uso desse ambiente para apoiar o processo de engenharia reversa e reengenharia, como pode ser visto na figura 5.

Um sistema legado exemplo terá sua interface substituída por uma interface OO com RV gerada com apoio do ambiente GaCIV, conforme é mostrado na coluna “Código Legado” da figura 5. Em seguida, a versão segmentada, já existente, do mesmo sistema legado sofrerá o mesmo processo de reengenharia de interface, visto na coluna “Código Segmentado” da figura 5. Finalmente, como ilustra a última coluna da mesma figura, um novo sistema seguindo o paradigma de OO será gerado à partir do sistema existente. Esse novo sistema também terá sua interface reengenheirada, substituída por uma interface de quarta geração.

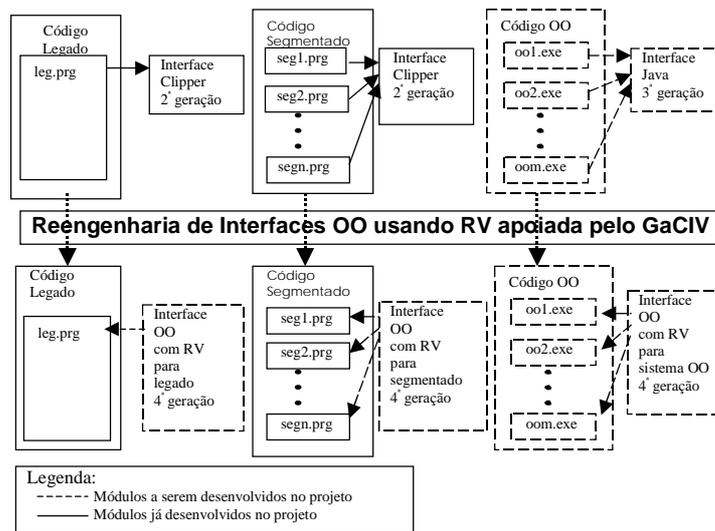


Figura 5 – Proposta de integração da engenharia reversa com RV

5. Estudo de Caso – O Sistema de Administração de Oficina Mecânica

O sistema exemplo a ser utilizado neste estudo é o mesmo apresentado em [18], que exibiu o processo de engenharia reversa e segmentação de um sistema legado. Ele foi originalmente desenvolvido em Clipper 5.0, com aproximadamente 20K linhas de código, com 25 tabelas do tipo “dbf”, relacionadas por chaves estrangeiras. Trata-se de um sistema real de uma oficina mecânica que controla os serviços executados em veículos, bem como o controle de estoque das peças utilizadas e de sua reposição.

O cliente chega à mecânica e solicita serviços para o seu veículo. Um cliente pode ter vários veículos. O mesmo veículo pode retornar à mecânica várias vezes, uma ordem de serviço específica é preparada para cada visita. Ela contém dados sobre o cliente, veículo e serviços a realizar. Quando o serviço for completado, a ordem de serviço do cliente é completada com relação às peças usadas e a mão-de-obra realizada. Algumas peças podem não existir no estoque e devem ser adquiridas fora, mas também são inseridas na ordem de serviço. O modelo de veículo deve ser registrado no sistema para que tabelas de preços sejam utilizadas para o pagamento dos serviços elétricos ou mecânicos, de acordo com o modelo do veículo.

O sistema legado apresenta uma estrutura de interface de pouca profundidade, isto é, a partir das diversas opções apresentadas na tela principal pode-se, em geral, acessar diretamente uma operação do sistema. Essa organização não permite a representação de níveis de abstração da aplicação com recursos de RV durante a reengenharia de interface.

Ao contrário do sistema legado, o sistema segmentado apresenta uma estrutura de interface profunda. Assim, permite a escolha de diferentes níveis de abstração para a aplicação da nova interface com recursos de RV. Essa estruturação será explorada neste trabalho, a fim de avaliar comparativamente a usabilidade das interfaces textuais versus interfaces com recursos de RV. O estilo de interface do sistema legado e segmentado, implementados em Clipper, pode ser representado pela tela ilustrada na figura 6.

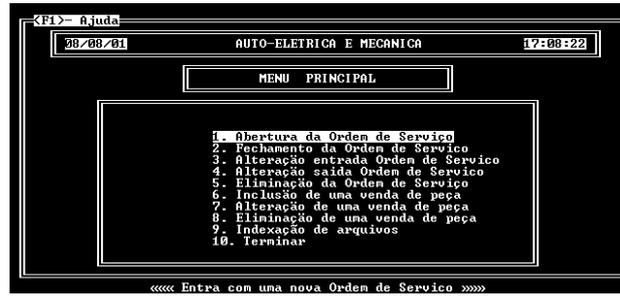


Figura 6 – Interface legada do sistema exemplo

6. Resultados e Discussão

Inicialmente foi realizado um estudo sobre as técnicas de desenvolvimento de interfaces como base de conhecimento para a realização das atividades de engenharia avante de interfaces do projeto. Esse estudo visou a abrangência das principais abordagens aplicadas nessa atividade: Modelagem Cognitiva, Projeto Centrado no Usuário e Projeto Participatório. Foram considerados diversos aspectos, tais como: possibilidade de aplicação no desenvolvimento de OOUT's, facilidades oferecidas, complexidade do uso, notações utilizadas, etc. Os métodos, metodologias e ciclos de vidas estudados foram: GOMS [4] (*Goals, Operators, Methods and Rules Selections*), LUCID (*Logical User Centered Interaction Design*) [5], USE (*User Software Engineering*) [25], *The Bridge* [8], OOHDM (*Object Oriented Hypermedia Design Method*) [9], construção de interfaces utilizando *Statecharts* [11], Engenharia de Usabilidade [17] e OVID (*Object, View and Interaction Design*) [20]. De acordo com suas características direcionadas ao desenvolvimento de interfaces OO, a utilização da metodologia OVID será priorizada neste projeto.

O projeto encontra-se em fase preliminar de construção do gabarito para composição das interfaces. O entendimento do sistema legado e segmentado já foi concluído, sendo que está em andamento a composição do gabarito utilizando o GaCIV e modelos 3D disponibilizados na Internet, com objetos relevantes para o desenvolvimento da interface de um sistema de automatização de uma oficina mecânica.

Com os estudos e a compreensão das funcionalidades do sistema exemplo, percebeu-se que os objetos do gabarito devem se adequar aos usuários do sistema, representados por funcionários da oficina, exigindo assim, objetos que representem metáforas mais significativas dos serviços de gerenciamento da oficina, além das metáforas relevantes a serviços oferecidos diretamente a clientes. Percebeu-se, então, a necessidade de objetos que representem atividades abstratas de maior complexidade para representação gráfica. O gabarito em edição foi utilizado na construção da interface ilustrada na figura 1.

Devido à possibilidade de escolha de diferentes níveis de abstração para a aplicação da nova interface com recursos de RV no sistema segmentado, a abordagem da reengenharia

da interface desse sistema será realizada de duas formas diferentes com o objetivo de uma análise comparativa. A primeira, é substituir toda a profundidade de um ramo da estrutura da interface, atingindo os níveis mais baixos de abstração possíveis, por objetos de realidade virtual. A segunda forma é substituir a interface legada em um nível intermediário de abstração. Dessa forma pode-se verificar a eficiência do uso da interface com recursos de RV em diversos níveis de abstração, com relação aos aspectos de usabilidade.

7. Considerações Finais

Os resultados esperados deste trabalho, em fase inicial de desenvolvimento, incluem a validação do paradigma de OO no processo de desenvolvimento de interfaces avançadas com recursos de RV, dentro do contexto da Engenharia Reversa e Reengenharia. Também espera-se que o ambiente GaCIV seja confirmado como uma ferramenta de apoio efetivo ao desenvolvedor de interfaces e de sistemas em suas tarefas junto ao usuário final, seja em um processo convencional de Engenharia de sistemas ou em um processo de Reengenharia, para a aplicação da RV no desenvolvimento de interfaces avançadas. Espera-se, assim, comprovar que o processo de reengenharia de interfaces com apoio do ambiente GaCIV se torna mais eficiente se o sistema em questão foi desenvolvido sob o paradigma OO.

Diretivas para a reengenharia de interfaces com recursos de RV, utilizando o GaCIV como ferramenta de apoio, é outro resultado esperado.

Deverão ser verificadas as vantagens e desvantagens em integrar uma nova interface com recursos de RV através do ambiente computacional GaCIV com os sistemas; 1) legados que possuem documentação obtida por um processo de engenharia reversa OO; 2) segmentados que têm o código implementado na linguagem procedural com características de OO inseridas a partir dos modelos obtidos com a engenharia reversa OO; 3) submetidos ao processo de reengenharia com a mudança de paradigma de desenvolvimento como de linguagem de programação OO.

8. Agradecimentos

Os autores agradecem à FAPESP (processo número 00/04577-0) o apoio financeiro dado a este trabalho.

9. Referências

- [1] ASSIS, A. S. F. R.; SILVA, J. A. C. Development of Interactive Systems with Virtual Reality Based on User Centered Design. In SCI 2000 - World Multiconference on Systemics, Cybernetics and Informatics, 4, 2000, Estados Unidos. Proceedings, julho/2000.
- [2] ASSIS, A. S. F. R.; SILVA, J. C. A. Meeting the Challenge of Systems Development with Virtual Interfaces. In: 2000 International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges (IS2000). Proceedings. Japão, Aizu-Wakamatsu, Fukushima. Novembro 5-8, 2000. p. 562-566.
- [3] BROWN, J. Methodologies for the Creation of Interactive Software. Victoria University of Wellington, Nova Zelândia, 1996.
- [4] CARD, S. M., T. & NEWELL A., The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1983.
- [5] COGNETICS. LUCID, URL: <http://www.cognetics.com/lucid/index.html>. Consultado em Dezembro/2000.

- [6] COLLINS, D. Designing Object-Oriented User Interfaces Estados Unidos: The Benjamin/Cummings Publishing Company, Inc 1995.
- [7] CONSTANTINE, L. *Essential Modeling - Use Cases for User Interfaces*. Interactions ACM, abril 1995, p. 34-46.
- [8] DAYTON, T. et al. Bridging User Needs to Object Oriented GUI Prototype via Task Object Design, User Interface Design – Bridging the Gap from User Requirements to Design, Estados Unidos: CRC Press LLC, 1998.
- [9] FELTRIM, V. D. et al. Uma modelagem do domínio de Engenharia Reversa de Software utilizando o método OOHDM. Notas do Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo – São Carlos, 1998.
- [10] GULLIKSEN, J. et al. User Centered Design - Problems and Possibilities: a summary of the 1998 PDC & CSCW workshop. SIGCHI Bulletin, v.31.n.2. abril/1999.
- [11] HORROCKS, I. Constructing the User Interface with Statecharts, Addison Wesley Longman Limited, England 1999.
- [12] IBM. Design Principles for Tomorrow. URL: http://www.ibm.com/ibm/hci/designer/uiarch/prinpl3_noframes.html. Consultado em maio/2000.
- [13] IBM. OOU Introduction. URL: http://www.ibm.com/ibm/hci/guidelines/design/ouui_introduction.html. Consultado em maio/2000.
- [14] KUHN, S. & MULLER, M. J. Participatory Design. Communications of the ACM, v.36, n.4, p.25-28, junho/1993.
- [15] LUCENA, F. N. & LIESENBERG, H. K. E. Interfaces Homem-Computador: Uma Primeira Introdução. URL: <http://www.dcc.unicamp.br/proj-xchart/start/welcome.html>, 1997. Consultado em abril/1999.
- [16] MORAN, T. The Command Language Grammar: A Representation for the User Interface of Interactive Computer Systems, International Journal of Man-Machine Studies, vol. 15, nro. 1, pg. 3-50, 1981.
- [17] NIELSEN, J. Usability Engineering. Estados Unidos: AP Professional, 1993.
- [18] PENTEADO, R. et al Improving the Quality of Legacy Code by Reverse Engineering . In Proceedings of ISAS'98 – 4th International Conference on Information Systems, Analysis and Synthesis. Orlando, Florida, pp 364-370 – Vol. I, julho/1998.
- [19] REISNER, P. Formal Grammar and Human Factors Design of na Interactive Graphics System, IEEE Transaction on Software Engineering, vol. SE-7, nro. 2, pg. 229-240, 1981.
- [20] ROBERTS, Dave et al. Designing for the User with OVID: Bridging User Interface Design and Software Engineering, Estados Unidos: Macmillan Technical Publishing, 1998.
- [21] SILVA, J. C. A. A Realidade Virtual e a Qualidade de Interfaces. In: workshop de qualidade de software - XII Simpósio Brasileiro de Engenharia de Software , Maringá, 1998. Anais. Maringá, PR, 1998, p. 11-17.
- [22] SILVA, J. C. A. Development of Virtual Interfaces Using Configurable Templates. In: ICCIMA99 – 3RD International Conference on Computational Intelligence And Multimedia Applications, New Delhi, India, 23-26 September 1999. Proceedings. India, 1999, p.354-358.
- [23] SILVA, J. C. A & PENTEADO, R. A. D. A Study of the Integration of Reverse Engineering and Interfaces with Virtual Reality. In: SCI2000 and ISAS2000 – 4th World Multiconference on Systemics, Cybernetics and Informatics and the 6th Int. Conf. on Information Systems Analysis and Synthesis, Orlando, USA, July 23-26, 2000 Proceedings. USA, 2000, p. 112-117.
- [24] VAN DAM, Adrian. Post-WIMP User Interfaces. Communications of the ACM, v.40, n.2, p.63-67, fevereiro/1997.
- [25] WASSERMAN, Anthony I. et al. Developing Interactive Information Systems with the User Software Engineering Methodology. IEEE Transactions on Software Engineering, v.12, n.2, p.326-345, fevereiro/1986.
- [26] WEISS, Patrice & JESSEL, Adam S. Virtual Reality Applications to Work, URL: <http://www.infocommons.utoronto.ca/atrc/rd/library/papers/weiss.html>
- [27] WOOD, E. Larry, User Interface Design – Bridging the Gap from User Requirements to Design, Estados Unidos: CRC Press LLC, 1998.

Semiotic Considerations on Direct Concept Manipulation as a Distinct Interface Style for Learnware

Clarisse Sieckenius de Souza¹, Kamran Sedig²

¹Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
22453-900 - Rio de Janeiro, RJ – BRAZIL

Information and Media Studies & Computer Science
The University of Western Ontario
Middlesex College, Rm 360
London, Ontario, CANADA, N6A 5B7
clarisse@inf.puc-rio.br, sedig@uwo.ca

Abstract. *Although direct manipulation (DM) has been considered a potentially inappropriate interface style for learnware, convincing indication that the problem with DM lies rather on “what is manipulated” than on the “manipulation” itself has been obtained from empirical studies in the domain of Euclidean geometry. A semiotic examination of such findings suggests that there may be theoretically-based constraints on the kinds of visualizations that yield appropriate direct concept manipulation interfaces for learnware.*

Introduction

Previous research about interface styles for learnware has suggested that direct manipulation (DM) is a potentially inappropriate design choice in this case, because ease of use and low cognitive effort may not be effective for learning purposes [Golightly, 1996; Golightly et al., 1996; Golightly & Gilmore, 1997; Holst, 1996; Rappin et al. 1997]. Convincing indication that the problem with DM lies rather on *what is manipulated* than on the *manipulation* itself has appeared in empirical studies carried out with apprentices using Super Tangrams[®] [Sedig et al., 2001] to learn and retain the knowledge involved in translation, rotation, and reflection of geometric objects. Direct Object Manipulation (DOM) has been shown to lead to a faulty understanding of abstract general concepts in the domain of 2-dimensional transformations in Euclidean geometry, whereas Direct Concept Manipulation (DCM) has led apprentices to learn the targeted concepts.

The reification of abstract concepts as visual manipulable forms hinges on important semiotic issues. When it comes to designing DCM interfaces for learnware, the availability of a general framework to guide choices among visual icons, indices or symbols would possibly increase the quality of such applications and their effectiveness as educational tools. Nevertheless, practical design guidelines and techniques can only emerge from a systematic and comprehensive analysis of visual communication codes and their impact on learning.

This paper is a first step in this direction. It presents a semiotic analysis of the Super Tangrams[®] interface style, whose dimensions are used to reinterpret results reported in research about diagrammatic reasoning [Stenning & Inder, 1995; Stenning & Oberlander,

1995] and to conjecture about visualizations that could be used in learnware for a specific topic in Biology—Mendel's laws of dominance. The result of this study is a set of theoretical issues that, in the opinion of the authors, should be investigated in a long-term research agenda about the use of DM in learnware interface design.

Direct Concept Manipulation for Learning about Geometry

Super Tangrams® (ST) is a piece of learnware that has been implemented using different interface styles. ST is built around a puzzle-solving activity called tangrams. The concepts to be learned in this activity are three geometrical transformations: slide (or translation), turn (or rotation), and flip (or reflection). In ST, children solve tangrams puzzles by manipulating on-screen images of given geometric shapes. Three distinct styles of interaction — DOM (Direct Object Manipulation), DCM (Direct Concept Manipulation) and RDCM (Reflective Direct Concept Manipulation) — have been polished to give children an impression of a commercial game. This has been judged important to give experiments a realistic atmosphere and to prevent implementation discontinuities from interfering with the study.

In the DOM version, the user manipulates the geometric shapes directly. The desired outline is presented in the center of the screen, with the available pieces scattered around the periphery (see Figure 1). Buttons on the side allow the user to select flip (horizontally or vertically), rotate (clockwise or counterclockwise), or drag mode. The order of the puzzles is fixed and every puzzle has a fixed score, which is added to the users total score when the puzzle is completed. This keeps the learning activity in close association with game playing.



Figure 1: A snapshot of the DOM HCI style in ST

In the DCM version, the user directly manipulates visual representations of the transformation concepts rather than the shapes themselves. The screen has a coarse grid in the background and the idea of focussing on a piece (rather than on an operation) is introduced (see Figure 2). The user chooses a shape (shown by the piece having a different pattern), the operation (*Slide*, *Turn*, or *Flip*), and this causes the transformation representation and the ghost image of the shape to appear. As indicated in Figure 2, when *Turn* is selected the transformation representation is an arc attached to one vertex of the focussed shape. The arc has a moveable center and endpoint. Manipulating the center



Figure 3: A snapshot of the RDCM HCI style in ST

The main findings of empirical studies carried out using ST indicated that a shift from object manipulation to concept manipulation had a positive effect on the learning of geometrical concepts by 6th-grade children from 2 classes of an upper middle-class school in Canada (for details see previous publications by Sedig [Sedig, 1998; Sedig et al., 2001]). In particular, the DCM version was more effective than the DOM one and the RDCM version was most effective in promoting domain learning. Children interacting with the DOM version of ST were sometimes confused with translation, for instance, missing the point that this operation is achieved along a linear vector path and not along an arbitrary curvilinear one (which is typically observed in drag-drop interactions available in DOM). Differences in understanding between children from these two groups were also observed with respect to the concepts of rotation and reflection.

The relatively superior efficacy of the RDCM style for learning was attributed to the progressive fading of visual clues that served as scaffolding for the children's reasoning. In the absence of such clues, they had to make more inferences about the causal relations between on-screen manipulations and the desired final state of the operation. These relations should have been acquired in previous stages of the game, when the visual scaffolding was available.

These findings led to the conclusion that further investigations about the use of direct manipulation for learnware were necessary. Discarding DM as an inappropriate style altogether would have prevented these children from the experience of having access to the causal relations between variations on vector coordinates, angles and arcs, and certain patterns of spatial configurations of polygons in 2D space. A well designed mechanism for leading learners from visual to algebraic representations of abstract concepts thus seemed to have the potential to become a new metaphor for DM interfaces for learnware [Sedig et al., 2001]. However, some open issues could be immediately raised.

1. "what visual representations to use to facilitate development of proper conceptual models;
2. what elements of the representation of a concept should be allowed to be manipulated—i.e., what controls (or handles) to manipulate, how to embed these controls in the visual structure, and in what order, if necessary, they should be manipulated; and,

3. what type of mouse interaction protocol to implement to direct learners attention towards the essential aspects of a concept.” [Sedig et al., 2001, p. 55].

The next section will advance some steps in the direction of answering the above questions, resorting to a semiotic analysis of the ST interface.

A Semiotic Account of the Super Tangrams' Interface

The primary theoretical reference of this analysis will be a number of sign classifications proposed by Peirce [Peirce, 1931-58]. Consolidated accounts of the main aspects of Peircean Semiotics [Nöth, 1998; Danesi & Perron, 1999] have been used to organize our analysis and to lead us more effectively to some interesting conclusions.

HCI literature does not provide as many instances of sound semiotic accounts as the generalized use of such words as *icons*, *indices* and *symbols* would lead us to believe. In contemporary Semiotics, these terms have technical meanings that originate from one of the numerous dimensions of sign classifications proposed by Peirce, namely that which relates different possibilities of representations (*representamina*) to three universal phenomenological categories (*firstness*, *secondness*, and *thirdness*). Peirce proposed that these radical categories could be used to organize all human experience in the world.

Firstness is the category of immediate perception that seizes the presence of phenomena, but does not relate them to anything else. An example of firstness is the perceptual quality of *redness* that emanates from whatever is red. *Secondness* is the category of association between two phenomena. Examples of secondness are the (causal) association between rain and wetness and the (contiguity) association between roads and vehicles. *Thirdness* is the category of chained associations, among three or more different phenomena. An example of thirdness is the (inferential) association that binds the proverbial mortality of Socrates with his being human and with humans being mortal, found in most introductory lessons to deductive reasoning. However, thirdness may occur in non-syllogistic chains of associations such as is the case with the relation between cars, roads and traveling, for example.

Before we proceed with semiotic definitions, we should go back to our theme and examine DCM and RDCM interfaces as implemented in ST. We are led to conjecture that appropriate visual representations of domain concepts should typically figure in associations that fall in the category of thirdness (or, for the sake of simplification, argumental chains of association). Should they fail to do so, and yield only associations of secondness (or, once again for the sake of simplification, mere relatedness), the progressive path from visual to algebraic concept formulations would be impossible to achieve. The lack of correspondence between a pair of phenomena (say the current location of a polygon and a given trajectory of the mouse on screen) and a third one (in this case the length and direction of a vector originating in the polygon) is precisely what has been reported to be missing in the learning of children who used the DOM interface.

The three Peircean categories have been used in their day to build a complex philosophical framework, within which Peirce investigated even the most basic principles of scientific discovery. Therefore, applying them to the analysis of signs present in different stages of cognitive or epistemic evolution is not an original endeavor. It leads us to interesting realizations about the history of scientific knowledge, as in the case of Mendel's findings about the principles of heredity [Dennet, 1995]. The path that led Mendel from his

observations about crossing different species of peas and the laws of dominance and segregation, for instance, can be accounted for as a progression from firstness to thirdness. At first, the occurrence of tall- and short-stem peas, of green and yellowish color, struck him as having a potential meaning (firstness). The search for meaning led him to probe the possible relations between different types of peas when crossed with each other. The recurring patterns derived from observable invariant features in the crossings emerged as systematic heredity relations (secondness) that applied to the realm of peas. But only when the association of these phenomena with such abstract meanings as the notion of dominance and segregation emerged was Mendel able to formulate his major contribution to the field of Genetics. The category of thirdness was achieved when the principles gained autonomy from experimental observations and entered the realm of abstract conceptualizations and signification.

The basic Peircean categories have also been used to generate further classifications and distinctions for whatever it is that can be taken by anybody to mean something else—i.e. classifications of signs. The most popular classification known to-date is that of icons, indices and symbols. Icons are a class of signs that relate to the firstness of their object (or referent). For this reason, they are often said to *resemble* their object (e.g. ♥ and 'heart'), although there is much more involved in iconicity than this simple idea might lead to suppose (see [Eco, 1976] for an extensive discussion). Indices are a class of signs that relate to the secondness of their objects. They typically relate to causality, spatiality or temporality [Nóth, 1998], as is the case of smoke taken as a sign of fire (causality), road as a sign of vehicles (spatiality), or thunder as a sign of rain (temporality). Finally, symbols relate to the thirdness of their object, by virtue of an arbitrary convention or law. They have the power of denoting objects without any relation to their firstness or secondness. Most verbal discourse signs are symbols *par excellence*.

Once again returning to the central theme of this paper, the complete cycle of learning geometric concepts [Sedig et al., 2001] should lead to algebraic formulations of the principles learned with RDCM. In other words, children should be able to verbalize (or formulate *linguistically*) the underlying general principles of translation, rotation and reflection. This amounts to mastering the appropriate symbols used for representing these concepts. Notice that an examination of the nature of signs appearing in the DOM style interface is bounded by the fact that only icons and indices are available on screen. The existing symbols (labels on the buttons to be pressed) will exhibit causal relations with the interactive affordances that do not, however, explicitly include signs associated to axes, angles and arcs. In contrast, the DCM style interface includes visual signs for these notions. Most of these visual signs are originally indexical, in that they relate to the secondness of the phenomena they refer to. Such is the case of vectors and arcs. Both exhibit visual properties that are contiguous with the dimensions of the represented concepts. But they also have symbolic equivalents that are expected to arise in the abstract formulation the children should produce to solve problems in tests and quizzes, for example. To illustrate this point, we can refer back to Figures 2 and 3, in the previous section, and verify that the manipulable objects all have *names* (which we must of course use even to describe what is in the picture): 'arc of rotation', 'line of reflection', 'coordinates', and so on. The availability of such *named* visualizations is precisely what provides the progressively fading scaffolding path that appears in the RDCM version of the interface. Learning the names of concepts, and being able to substitute names for visible

(and *tangible*) occurrences of concepts is what characterizes the ability to participate in discourse.

We will not engage in further considerations about the parallel between such semiotic characterization of sign-mastering progression, on the one hand, and learning, on the other. Our goal is only to discuss HCI design issues raised by previous research on DM and learnware. However, an interesting example (among many others) of the connections between Semiotics and Cognitive Psychology is Piaget's genetic epistemology [Piaget, 1970]. A parallel between Peirce's phenomenological categories and Piaget's levels of cognitive evolution can be easily drawn. Moreover, Lakoff's extensive account of philosophical, psychological and linguistic research that supports his views about the pervasive role of phenomenologically-rooted metaphors in human discourse [Lakoff & Johnson, 1980; Lakoff, 1987] provides additional evidence of how the linguistic rendition of abstract concepts is grounded in relations and perceptions that we could characterize as cases of secondness and firstness in human experience.

Thus, our semiotic account of the findings reported in experiments carried out with Super Tangrams[®] allows us to propose that:

- DOM is not an appropriate interface style for learnware if the objects being manipulated are not sufficient to represent the necessary denotations involved in linguistic formulations of the principles to be learned.

In other words, if we expect learners using DOM to formulate principles such as "when crossing parents with pure contrasting heredity traits, the next generation of individuals will exhibit the characteristic of only one of the contrasting traits", we should provide them with visual signs (static or dynamic) that can stand for such notions as 'crossing', 'parent', 'heredity trait', 'pure', 'contrasting', 'next generation individuals', 'characteristic', 'only one', and 'exhibit'. Note that the visual code is made up of visual representations and manipulations of these.

- The evolution from manipulation experiences to linguistic formulations leads us to integrate DCM and RDCM into a two-stage single interactive style. This new style may be more appropriate for learnware if (a) the concepts being manipulated are sufficient to represent the necessary denotations involved in linguistic formulations of the principles to be learned, and (b) the constructive path leading from secondness to thirdness of signs used to represent the concepts involved in learning can be traced with intermediary visual signs that are progressively less indexical and more symbolic.

If there are no intermediary signs, leading from indices to symbols, in the representations used by the interface designer, an important step in constructive learning may be missing. The ST experiments showed that the three levels of progressive fading in RDCM played an important role in leading children to master the geometric concepts underlying translation, rotation and reflection of polygons. If, for example, in the domain of Mendelian heredity laws, there is not a representational scaffolding that will help learners generalize from concrete crossings to abstract systematic principles, this step will fall outside the scope of the specific learnware tools. Learners may eventually reach the generalization stage, but the progression will not be modeled in the system as a *path*. It will be a discrete

change from one stage to another. The *path*, as in ST, requires a set of representations that are increasingly more linguistic (and inferential) and less visual.

This observation also suggests that DCM without RDCM may prevent learners from actually reaching the desired stages of knowledge embedded in learnware. Nevertheless, DCM and RDCM stand out as valuable *stages* of concept manipulation in the ST interface style.

Euler Circles and Punnet Squares

The use of images and diagrams as teaching aids is pervasive. In typical classroom situations, they are used as visualizations of specific aspects of contents being taught. Some visualizations can be easily used to elicit the correct responses in exercises and tests, whereas others can be used in cartoon-style sequences to depict different stages of transformations. Some are more concrete, and some more abstract. Some are quite conventional, and others are the products of a teacher's imagination and talent. Therefore, saying that direct manipulation interfaces are bad for learnware is counter-intuitive. However, "graphics are frequently used in introductory teaching of abstract subjects, but are abandoned by students as they gain proficiency" [Stenning & Inder, 1995, p. 318]. Two examples of how diagrams can be used to support the teaching and learning of different topics illustrate some of the issues DCM and RDCM designers should be prepared to face.

Euler Circles are a common graphical representation used in teaching syllogistic reasoning. For instance, the expression "all humans are breast-fed and some animals are not breast-fed; therefore some animals are not human" is logically correct, although pragmatically strange (since our knowledge of the world tells us that *no one* animal is human). If Logic is taught based on a linguistic medium, the learning may be impaired by the interference of such factors as the one just exemplified. However, using circles (proposed by Euler in the XVIII century) can considerably improve learnability. Graphical representations of abstractions relative to quantification are more easily grasped than in the linguistic form (see Figure 4, adapted from Wang et al., 1995). The advantage of the diagram is that common-sense information that inevitably plays a role in linguistic formulation (where predication is interpreted as a cognitive category) is omitted in favor of more salient graphical properties (i.e. intersection and inclusion of circles). Therefore, if the idea is to lead learners to become proficient in interpreting and evaluating expressions such as:

$$((\forall x H(x) \rightarrow Bf(x)) \wedge (\exists y A(y) \wedge \neg Bf(y))) \rightarrow (\exists y A(y) \wedge \neg H(y))$$

the diagrammatic representation is helpful as a mechanism to separate logic predication and propositional calculus from cognitive interpretations of the world and psychologically realistic information processing.

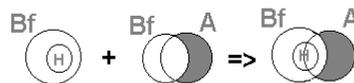


Figure 4: Euler Circles representing the expression "all humans are breast-fed and some animals are not breast-fed; therefore some animals are not human"

However, diagrammatic reasoning does pose problems. The interpretation of graphical constants is always affected by the emergence of cognizable elements that participate in the

construction of the diagram [Wang et al., 1995]. A closer examination of Figure 4 might give elements to support the interpretation that some H are A, which is not a valid logic conclusion. An attempt to correct this representation by removing the intersection between circle H and circle A, as shown in Figure 5, would also lead to interpretations that are logically incorrect (i.e. that no A is H).

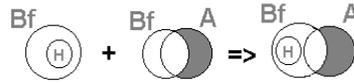


Figure 5: An alternative representation for “all humans are breast-fed and some animals are not breast-fed; therefore some animals are not human”

This situation has motivated research in different directions. Wang et al. (1995) have proposed a series of rules and constraints for constructing good diagrammatic representations. Stenning and Inder (1995) and Stenning and Oberlander (1995) have chosen another path and explored modes, media, and expressiveness of representation systems. The latter have proposed a hierarchy of representation systems (minimal abstraction systems, limited abstraction systems, and unlimited abstraction systems) and have provided experimental cognitive evidence that limited abstraction representation systems (LARS), which allow for representing *indeterminacy* are better as cognitive scaffolding in learning than minimal (MARS) and unlimited abstraction representation systems (UARS). It is interesting to notice that the problem with either Figures 4 or 5 is their excessive degree of (emergent) determination with respect to the relation between H and A.

Stenning and Inder (1995) have also analyzed matrices and tables as graphic representations. They have concluded that these kinds of visual aids can only represent data that is fully specified on the dimensions present in the matrix. Partially determined data cannot be represented, for the assignment of data to a cell automatically specifies its value in all of the table's dimensions. This observation has an interesting consequence for teaching Mendel's laws of heredity with the use of Punnet Squares (see Figure 6).

If we take the Law of Dominance, for example, we can use the Punnet Square to represent the principle that “when crossing parents with pure contrasting heredity traits, the next generation of individuals will exhibit the characteristic of only one of the contrasting traits”. The Punnet Square is a matrix (a table) where the traits of the female and male parents are displayed in the first column and first row. The possibilities of genetic combinations are displayed in the cells of the table, by the concatenation of codes used to represent each heredity trait. Figure 6 shows a Punnet Square for the crossing of two pure types: tall-stem pea (TT) and a short-stem pea (tt). The principle is represented by the fact that all genetic combinations of the offspring are Tt. Observably, this corresponds to the fact that the individuals show the characteristics of trait ‘T’, despite the presence of trait ‘t’ in their chromosomes.

	T	T
t	Tt	Tt
t	Tt	Tt

Figure 6: The Punnet Square representing Mendel's Law of Dominance

The interesting fact about Punnet Squares, from Stenning and Inder's perspective, is the emerging meaning in the representation, that no genetic traits can be indeterminate. Therefore, the Punnet Square does not easily yield itself to a continuous constructive progression of the concept. The table can only be filled out when and if the principle is fully known. It cannot be used in the process of learning the principle (where knowledge goes through successive stages of decreasing indeterminacy).

These two brief examples from different domains provide us with some interesting insights regarding DCM and RDCM for interacting with learnware. They draw attention to the fact that the visual representation of indeterminacy requires diagrams or images that can represent *types* [Stenning & Inder, 1995; Stenning & Oberlander, 1995]. Indeterminacy may be important for different reasons. First, it may be the case that there are cases of indeterminacy in the domain of knowledge being learned (see the Euler Circles example). Second, it may alternatively happen that, although there are no cases of indeterminacy in the domain, the gradual construction of knowledge structures may require the representation of *indeterminate* states in the learners' conceptualizations (see the Punnet Square example).

Viewed from this perspective, the evolution from DCM to RDCM interaction with Super Tangrams[®] corresponds to a gradual decrease of the referential determinacy of visual representations until a minimal threshold is reached. Visual cases of indeterminacy, in ST, are compensated with internalized concepts that are arguably available in symbolic form (the pre-linguistic mental formulation of general principles that have been learned from previous more determinate stages of representation). We can then anticipate that the style of learning adopted in ST would not be easily reproducible for teaching Logic with Euler Circles or Genetics with Punnet Squares.

Conclusion

The previous sections give us some interesting elements to respond to the issues posed so far about the value of DCM and RDCM as distinct interface styles in learnware. Firstly, in semiotic terms DCM and RDCM can be argued to be more closely related than has been originally suggested. In fact, the phenomenological categories of *secondness* (that refers to establishing relations between two entities) and *thirdness* (that refers to establishing relations between at least three entities) can suitably characterize the interface signs used in Super Tangrams[®]. The relevance of this argument is that the availability of a theoretical framework that can adequately account for design choices based on heuristic guidelines and experimental observations allows us to explore a number of predictions and explanations of HCI phenomena in the domain of learnware. These predictions are backed by semiotic theory, and can help us not only probe the spectrum of possible answers to known questions, but perhaps more importantly to formulate new questions and advance our knowledge in the field.

To illustrate this point, we can already conjecture about the issues raised in previous studies with the system.

1. *What visual representations [should we] use to facilitate [the] development of proper conceptual models?*

Our analysis seems to support the conjecture that we should use those representations (a) that can be related to the secondness as well as thirdness of their objects, and (b) that can

themselves be produced in iconic or indexical form as well as in symbolic form. This condition is apparently necessary to ensure a *continuum* from DCM into RDCM interaction.

2. *What elements of the representation of a concept should be allowed to be manipulated —i.e., what controls (or handles) to manipulate, how to embed these controls in the visual structure, and in what order, if necessary, [should] they be manipulated?*

Visual codes include static and dynamic forms. Static forms are visualizations; dynamic forms are manipulations. So, a concept must not necessarily be represented by static forms alone — it can be represented by a manipulation of visual forms. The decision on how to express the possibility of manipulation (typically by the use of a handle) is secondary to the decision of which kind of manipulation pattern to use. An adequate integration (or embedding) of signs of interactive possibilities into signs of domain concepts may not be an easy task.

For instance, if learners are expected to acquire the appropriate grammatical knowledge for producing embedded sentences with the pronoun *whose* in English, the use of DCM and RDCM interfaces is a major design challenge. Although advanced grammatical studies in Linguistics eventually lead learners to represent syntactic structures by means of trees and graphs, at the time this knowledge is taught to students the use of trees and graphs is excessively abstract. The alternative teaching aid is often one of resorting to metaphors, like bounding boxes around phrases, arrows directing the movement of syntactic constituents, and colored overwriting to indicate pronominal substitutions (see Figure 7).

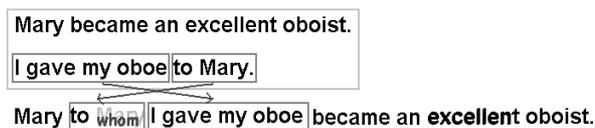


Figure 7: Visualizing embeddings of wh- clauses in English

Thus, when building a DCM-RDCM interface for teaching a subject matter like this, a designer should select some metaphoric expressions for the domain concepts and iteratively seek for an adequate means to express interactive patterns that can be used with them. The cost of metaphoric expressions is a bigger cognitive effort required from learners, who will have to bridge the gap from metaphoric to abstract concept formulation without the help of the system. In the example above, learners should go from moving boxes and replacing constituents with pronouns to acquiring general extraposition and pronominalization principles for embedding *wh-* clauses into main sentences. Likewise, the order of manipulation should follow a path from narrower to wider gaps, supporting learners in the process of gaining increasing competence in producing abstract discourse about the general principles of the domain. In this respect, Lakoff's theory of metaphor and cognitive ontology should be a resourceful instrument for supporting design decisions.

3. *What type of mouse interaction protocol [should we] implement to direct learners attention towards the essential aspects of a concept?*

A semiotic revision of DCM and RDCM styles of interface for learnware seems to indicate that this issue cannot be treated independently of the former. In fact, mouse interaction protocols are part of the dynamic signs that will constitute the representation of concepts.

Treating this issue as an independent factor is not likely to produce results that will directly impact the quality of DM for learnware. Some peripheral affordances could be discovered along this sort of study, however, providing designers with a wider range of choices of non-metaphoric domain concept representations. Yet, the value of such affordances as learning scaffolding should be subordinate to the value of signs themselves as carriers of domain meanings.

Finally, a theoretical approach of the kind we propose in this paper indicates that a tension between traditional graphic representations used in different domains (such as Euler Circles, Punnet Squares, syntactic trees and metaphors thereof), on one side, and concept visualizations that are tractable for smooth DCM-to-RDCM transitions, on the other, will probably arise. Because traditional visualizations were created to be used in non-interactive *media* such as paper and blackboards, *manipulation* was never an issue with them. So, it may well be the case that such traditional resources offer poor manipulation patterns in interactive *media* such as the computer. By the same token, more easily manipulable visualizations may involve unnecessarily complex metaphoric stretches in visual sense-making, if considered from a strictly pedagogical point of view.

As a conclusion, we can say that a semiotic analysis of ST has given us one more degree of insight into the issues involved in DM style interfaces for learnware. Evidence from related work in diagrammatic reasoning and practices in teaching different subject matters suggests that we should continue with this line of investigation and carry out analytical experiments with other systems and interface styles. As soon as strong indications of the plausibility of these conjectures are obtained, the next steps toward a coherent body of theoretically-based design guidelines for HCI in the domain of learnware are: to formulate emerging design principles; to apply them in the redesign of existing systems (where the principles have not been observed); and to compare learners' performance with the original and the redesigned version of the application.

Acknowledgements

Both of us, authors, would like to thank the anonymous referees of this paper, who provided useful comments and suggestions for this improved version. We are indebted to the institutions and people who have made this work possible. Clarisse Sieckenius de Souza, in particular, would like to thank CNPq for supporting her research in many different ways. Special thanks go to Tom Carey, of LT3 at the University of Waterloo, Canada, who has insightfully encouraged us to explore the opportunities for cross-fertilizing our individual research works.

References

- [1] Danesi, M. and Perron, P. (1999) *Analyzing Culture. A Handbook and Introduction*. Bloomington, In. Indiana University press.
- [2] Dennet, D. C. (1995) *Darwin's dangerous idea*. New York. Simon & Schuster, Inc.
- [3] Eco, U. (1976) *A Theory of Semiotics*. Bloomington, In. Indiana University Press.
- [4] Forman, G., & Pufall, P. B. (1988). Constructivism in the computer age: A reconstructive epilogue. In G. Forman, & P. B. Pufall (Eds.), *Constructivism in the computer age*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- [5] Golightly, D. (1996). Harnessing the interface for domain learning. *Proceedings of CHI '96: Conference Companion*, 37-38.

- [6] Golightly, D., & Gilmore, D. (1997). Breaking the rules of direct manipulation. In S. Howard, J. Hammond, & G. Lindgaard (Eds.) *INTERACT '97. IFIP TC13 International Conference on Human-Computer Interaction*. London: Chapman Hall pp. 156-163.
- [7] Golightly, D., Gilmore, D., & Churchill, E. (1996). Puzzling interfaces: The relationship between manipulation and problem solving. *Adjunct Proceedings, HCI'96 Conference of the BCS HCI Group*, Imperial College London, August 1996.
- [8] Holst, S. J. (1996). Directing learner attention with manipulation styles. *Proceedings of CHI '96: Conference Companion*, 43-44.
- [9] Lakoff, G. (1987) *Women, fire and dangerous things*. Chicago, IL: The University of Chicago Press.
- [10] Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago, IL: The University of Chicago Press.
- [11] Nöth, W. (1998) *Panorama da Semiótica*. São Paulo. Annablume Editora.
- [12] Peirce, C.S. (1931-1958) *Collected Papers*. Volumes 1-8. Hartshorne & Weiss & Burk (organizers). Cambridge, Mass. Harvard University Press.
- [13] Piaget, J. (1970) *L'Épistémologie génétique*. Paris. Presses Universitaires de France.
- [14] Rappin, N., Guzdial, M., Realf, M., & Ludovice, P. (1997). Balancing usability and learning in an interface. *Proceedings of CHI '97*, 479-486.
- [15] Sedig, K. [or Sedighian, K.] (1998). *Interface style, flow, and reflective cognition: Issues in designing interactive multimedia mathematics learning environments for children*. Unpublished Ph.D. dissertation. Department of Computer Science, The University of British Columbia, Vancouver, Canada.
- [16] Sedig, K., Klawe, M., & Westrom, M. (2001) Role of Interface Manipulation Style and Scaffolding on Cognition and Concept Learning in Learnware. *ACM Transactions on Computer-Human Interaction*. No. 1, Vo. 8, pp. 34-59.
- [17] Stenning, K. & Inder, R. (1995). Applying semantic concepts to analyzing media and modalities. In J. Glasgow, N. H. Narayanan & B. Chandrasekaran (Eds.). *Diagrammatic reasoning: Cognitive and computational perspectives*. Menlo Park CA: AAAI & MIT Press.
- [18] Stenning, K., & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science*, 19(1):97-140.
- [19] Wang, D., Lee, J., & Zeevat, H. (1995). Reasoning with diagrammatic representations. In J. Glasgow, N. H. Narayanan & B. Chandrasekaran (Eds.). *Diagrammatic reasoning: Cognitive and computational perspectives*. Menlo Park CA: AAAI & MIT Press.

Um Modelo Conceitual para Programação por Usuários Finais

Sérgio Roberto P. da Silva¹, Clarisse Sieckenius de Souza²

¹Departamento de Informática – Universidade Estadual de Maringá (UEM)
Av. Colombo, 5790, zona 07 – 87020-900 – Maringá – PR – Brazil

²Departamento de Informática – Pontifícia Universidade Católica do Rio (PUC-Rio)
R. Marquês de São Vicente, 225, Gávea – 22453-900 – Rio de Janeiro – RJ – Brazil
srsilva@din.uem.br, clarisse@inf.puc-rio.br

***Abstract:** This paper describes a conceptual model for end-user programming. Based on Semiotic Engineering concepts we suggest that the task of 'using the software' can be viewed as a communication and representation process. To elicit the components involved in the task of creation of software extensions by end-users we employ the Jakobson's Verbal Communication Model. We also present the 'Abstractive Interpretation' and 'Semiotic Continuum' principles, which, according to our approach, should govern the organization of the codes involved in extensible software so that it could achieve good quality.*

***Resumo:** Neste artigo apresentamos um modelo conceitual para programação por usuários finais. Partindo de conceitos da Engenharia Semiótica, sugerimos que a tarefa de 'uso do software' possa ser vista como um processo de comunicação e representação. Para identificar os componentes envolvidos na tarefa de criação de extensões ao software pelos usuários finais, empregamos o Modelo de Comunicação Verbal de Jakobson. Apresentamos ainda os princípios da 'Abstração Interpretativa' e do 'Contínuo Semiótico' que, segundo nossa abordagem, devem reger a organização dos códigos atuantes no software extensível para que este seja de boa qualidade.*

1. Motivação

Uma tendência na indústria de software tem sido a de disponibilizar recursos para que os usuários finais possam configurar ou estender suas aplicações, numa tentativa de ampliar seu escopo de usabilidade e aplicabilidade [FISCHER '98] e, assim, satisfazer suas reais necessidades. Infelizmente, a simples disponibilidade de tais recursos não capacita estes usuários a estender suas aplicações pois, normalmente, falta-lhes conhecimento de como utilizá-los. Adler e Winograd [ADLER '92] propõem que o critério básico para a usabilidade de um software é a extensão do apoio que ele proporciona para que os usuários possam compreender e apreender o seu funcionamento, adaptá-lo e estendê-lo. A área de programação por usuários finais [CYPHER '93] [GOODELL '99] [LIEBERMAN '01] pode ser vista como uma resposta a tais anseios. Entretanto, que seja de nosso conhecimento, não existe um modelo teórico de consenso que apóie o desenvolvimento de aplicações extensíveis de forma satisfatória.

Muitos dos problemas enfrentados pelos usuários finais para a programação das aplicações extensíveis atuais são devidos às descontinuidades existentes entre sua interface e seu ambiente de extensão, mais especificamente entre as linguagens empregadas nestes dois ambientes [DE SOUZA '01]. Tais descontinuidades são causadas: 1) pela ausência de um mapeamento entre elementos empregados na linguagem de interface do software e na linguagem do seu ambiente de extensão ou vice-versa ou 2) pela ausência de clareza no mapeamento existente entre estas duas linguagens. Exemplos do primeiro caso podem ser reportados às situações em que a presença de um elemento de interface como, por exemplo, um ícone ou uma função de um menu, não tem sua expressão na linguagem do ambiente de extensão da aplicação ou vice-versa. Exemplos do segundo caso podem ser reportados às situações em que, apesar de o mapeamento entre as duas linguagens existir, não é possível ao usuário inferir diretamente quais os elementos de uma linguagem estão sendo mapeados em que elementos da outra linguagem.

2. O *framework* da Engenharia Semiótica

Em nosso trabalho na área de Interação Humano-Computador empregamos como referencial teórico a abordagem da Teoria Semiótica [ECO'76] [PEIRCE '31] e, mais especificamente, a Engenharia Semiótica [DE SOUZA '01] [DE SOUZA '93]. Na Engenharia Semiótica, o software é visto como uma **mensagem única e unidirecional de alto-nível** do designer de software para o usuário. Em vista disso, sua interface — o seu elemento de interação com o usuário — será considerada um **artefato de meta-comunicação** (uma mensagem constituída de forma intencional e regrada que pode gerar ou receber outras mensagens). Esta abordagem torna central o **aspecto comunicativo da interface do software** chamando-nos a analisá-lo mais cuidadosamente como um processo de comunicação e representação. Ela também nos traz novos *insights* para a dualidade entre designer e usuário final dentro do campo de aplicações extensíveis.

Para exemplificar este processo de comunicação, apresentamos na Figura 1 um esquema da nossa abordagem, em que o designer, o computador e o usuário exercem os papéis de emissor, de meio/canal de transmissão e de receptor, respectivamente. O designer, como emissor, criará uma mensagem — o software —, que será a concretização do significado pretendido da mensagem do designer — isto é, a sua compreensão do problema, a solução por ele proposta e as possibilidades de extensão desta solução por ele vislumbradas [DE SOUZA '96]. O conteúdo desta mensagem será decorrente de um modelo conceitual abstrato do software (proveniente da análise do domínio do problema, das classes de usuários e das tarefas que se pretende apoiar) e das decisões de design e de implementação tomadas pelo designer, resultando em um modelo de usabilidade projetado. Deste modo, ele representa o interpretante cristalizado do designer e, conforme Prates *et al.* [PRATES '00], conterá as *affordances* pretendidas pelo designer na linguagem de interface do software.

O usuário, por sua vez, através de seu contato direto com o domínio do problema, formará um modelo de usabilidade desejado para o software. Por outro lado, durante o processo de interação com a mensagem do designer — isto é, o software projetado — o usuário, por meio de sua interpretação dos elementos da interface deste software, atribuirá um significado a esta mensagem, criando assim o seu modelo de usabilidade percebido do software. Este é um modelo pessoal que representa o interpretante do usuário para a mensagem do designer. Ele contém as *affordances* percebidas pelo usuário, que são um

subconjunto das *affordances* pretendidas pelo designer. Assim, o significado atribuído por um usuário para o seu interpretante da mensagem do designer será uma variante (em diferentes graus de compatibilidade e consistência mútua) do significado atribuído por outros usuários, assim como daquele atribuído pelo designer. Para a comunicação ter sucesso, todas estas variantes deverão ser compatíveis e consistentes entre si (sobretudo compatíveis e consistentes com a variante do designer).

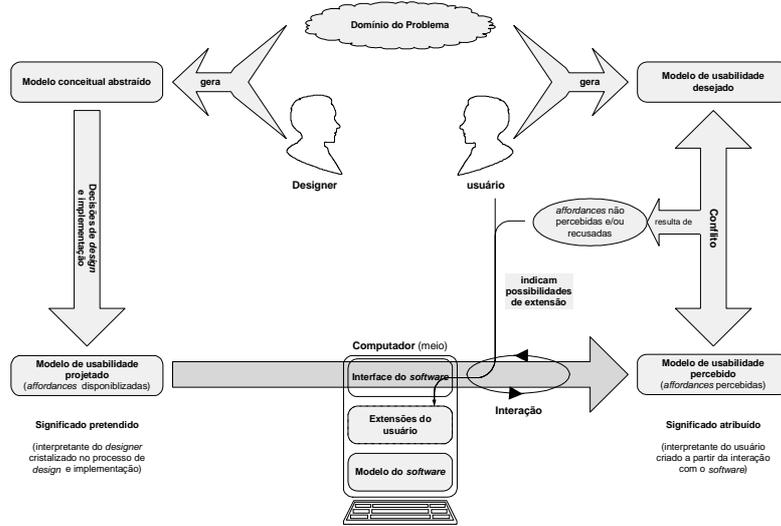


Figura 1: Framework de Engenharia Semiótica estendido para incluir a tarefa de End-User Programming (EUP).

Neste ponto, podem surgir problemas de usabilidade do software decorrentes dos conflitos entre o modelo de usabilidade desejado pelo usuário e o modelo de usabilidade percebido pelo usuário. Estes problemas, conforme descritos por Prates *et al.* [PRATES '00], poderão ser resultado da não percepção, pelos usuários, de *affordances* pretendidas pelo designer do software ou da recusa dos usuários em usá-las (por acharem que existe uma forma mais eficiente de executar uma determinada tarefa). De Souza [DE SOUZA '00] destaca que os problemas de comunicabilidade podem ocorrer em três níveis nos quais as *affordances* podem ser classificadas. São eles os níveis: 1) operacional — relativos às ações individuais realizadas pelos usuários; 2) tático — relativos à uma seqüência de ações que podem ser realizadas para alcançar determinados objetivos; e 3) estratégico — relativos às conceitualizações envolvidas na formulação dos problemas e no processo de solução de problemas.

A não percepção e a recusa dos usuários em usar as *affordances* pretendidas pelo designer indicam, por um lado, problemas no projeto do software porém, por outro lado, podem indicar também oportunidades de extensão ao software. Assim, por exemplo, no nível operacional, os usuários podem re-arranjar os signos disponibilizados pelo designer para enfatizar as *affordances* que são mais significativas para eles e, no nível tático, os

usuários podem criar métodos customizados que melhor satisfaçam suas necessidades de interação [DE SOUZA '00].

Desta forma, nos casos em que é possível resolver os conflitos, o usuário poderá tomar o papel de designer e acrescentar à aplicação novos elementos de funcionalidade por meio da criação de extensões. A fim de realizar esta tarefa, o usuário deverá conhecer o modelo da aplicação de forma a construir suas extensões de acordo com este modelo sem destruir a mensagem original do designer. Num cenário típico de uso de software extensível, haverá duas situações bastante distintas a serem consideradas: 1) o seu uso na realização de uma tarefa do usuário; e 2) o seu uso na criação de uma extensão 'a ele mesmo' pelo usuário.

3. O uso do software como um processo de comunicação

Para analisarmos a inter-relação entre os elementos participantes no processo de comunicação, constituído pelo uso do software, tomaremos como base teórica, além da Engenharia Semiótica, o Modelo de Comunicação Verbal de Jakobson [JAKOBSON '60]. Para Jakobson uma mensagem é transmitida de um emissor para um receptor. Mas, para que ela seja eficaz, é necessário que seja definido um contexto que seja apreensível pelo receptor ao qual ela se refere. Além disto, é necessário um código total ou parcialmente comum a ambos (este código é usado para transmitir as intenções e significados do emissor ao receptor) e um canal — isto é, um meio físico e uma conexão psicológica — que permita a eles entrarem e permanecerem em contato.

Aplicando este modelo a ambas as situações de uso de software extensível, identificamos que o canal de comunicação será sempre o hardware que executa o software, e seus periféricos. Este é um canal rico que constitui um meio multimodal o qual possibilita o emprego de múltiplos códigos na emissão e recepção de mensagens. Os demais elementos irão variar nas duas situações e serão analisados separadamente a seguir.

3.1 O uso normal do software extensível

No primeiro caso de uso de software extensível o emissor será o designer, uma vez que é ele quem cria o software — a mensagem original. O receptor será ora o usuário (sempre que ele interpreta as mensagens enviadas pelo preposto do designer), ora o próprio software (quando este tem de interpretar as mensagens enviadas pelo usuário por meio de comandos através de sua *UIL* — *User Interface Language*). O contexto será definido pelo domínio para o qual a aplicação é construída e pelo sistema computacional (o conjunto composto pelo sistema operacional e os elementos do seu sistema de interface padrão) no qual ele executará. O código empregado na criação das mensagens que serão trocadas entre a interface do software e o usuário será a *UIL* do software. É interessante observar que, pela abordagem Semiótica, o *background* do usuário também influenciará na interpretação das mensagens emitidas pela interface do software fazendo, portanto, parte do contexto.

A primeira observação a ser feita sobre este fenômeno é que este é um processo de comunicação em que o seu emissor — o designer do software — não está mais presente diretamente no processo comunicativo e, portanto, estamos lidando com uma mensagem na qual tudo o que era para ser dito pelo seu emissor já foi dito. Esta situação criará uma lacuna no processo comunicativo podendo gerar problemas à sua efetivação. Tais problemas poderão ocorrer em virtude do fato de que *a priori* o receptor ficará

impossibilitado de fazer uso completo da função metalingüística (ver definição a seguir), ficando limitado à leitura do *help* do software.

O mapeamento do uso de software extensível nos elementos do Modelo de Comunicação de Jakobson sozinho não nos traz grandes *insights*. Contudo, Jakobson também descreve as funções que cada um dos componentes de seu modelo exerce sobre a linguagem [JAKOBSON '60]. Estas funções são caracterizadas pelo enfoque dado a um dos componentes do modelo na mensagem e podem ser usadas tanto analiticamente, para ganhar *insights* no processo de avaliação da comunicabilidade da Interação Humano-Computador [PRATES '00], quanto sinteticamente para auxiliar na escolha dos elementos do código que melhor expressam a intenção comunicativa do designer durante a construção da mensagem. Neste trabalho discutiremos somente os efeitos das funções metalingüística e referencial sobre o software extensível, por serem estas as que mais diretamente afetam a tarefa de extensão do software.¹

A função metalingüística é a função da linguagem em que o enfoque está sobre o código empregado na comunicação e, portanto, ela está direcionada para a necessidade do receptor e/ou emissor verificar se estão usando o mesmo código. Assim, ela será empregada na elucidação de problemas de interpretação do código em que uma mensagem é transmitida. Logo, para um usuário de software, a perda ou limitação desta capacidade implicará a impossibilidade de esclarecer algumas dúvidas sobre a funcionalidade de elementos interativos da *UIL* que não lhe sejam diretamente interpretáveis (em função de seu conhecimento do domínio e/ou de sua alfabetização computacional e/ou da documentação do software).

É essencial observar que, devido à interface do software ser um artefato de meta-comunicação, a lacuna produzida pela ausência do designer a este processo comunicativo será preenchida pela sua criação — o próprio software — que, desta forma, o representará durante o diálogo metalingüístico com o usuário. Além disso, deve-se notar também que a *UIL* do software forma uma camada de abstração que oculta do usuário os demais níveis de linguagem necessários à comunicação com o ambiente computacional e o hardware da máquina. Portanto, para os usuários finais, **a *UIL* de um software representa o próprio software**, refletindo apenas os elementos comunicativos disponibilizados pelo seu designer. Por este motivo, durante o uso normal do software extensível, a maioria das dúvidas dos usuários recairão primordialmente sobre os elementos interativos da *UIL*, e a tarefa de esclarecê-las ficará a cargo da linguagem de explicação do usuário à *UEL* (*User Explanation Language*) — denominada também de linguagem de documentação e composta pelo *help online*, os *tooltips* e a documentação do software — que deverá conter as explicações necessárias ao usuário. A *UEL* não deve ser vista como um elemento exterior ao software, mas sim como um componente presente em todo software com a função primeira de representar o designer no diálogo metalingüístico. A Figura 2 apresenta a instanciação do modelo de Jakobson para o caso de uso normal de software.

É interessante observar que, em virtude da impossibilidade de inclusão de explicação para todas as dúvidas dos usuários de um software, a perda do uso natural da função metalingüística pelo usuário final será um problema sem solução total impondo,

¹ Uma análise da influência das demais funções e de seu uso na elaboração da Interação Humano-Computador pode ser encontrado em [da Silva '01].

desta forma, um peso bastante grande sobre a qualidade da *UEL* necessária para minimizar esta perda.

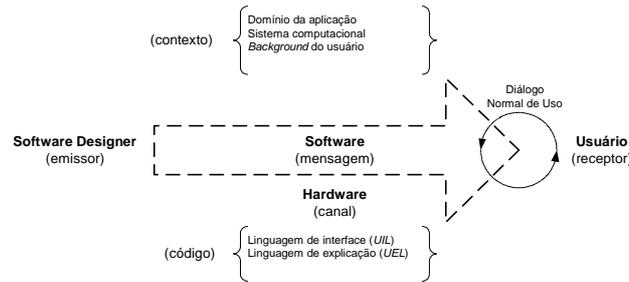


Figura 2: Modelo de comunicação de Jakobson completamente instanciado para o caso de uso normal do software extensível.

Jakobson também comenta que “todo processo de aprendizagem da linguagem (...) faz largo uso de operações metalingüísticas” [JAKOBSON '60], de modo que a impossibilidade de usar parte da função metalingüística afetará também a capacidade de aprendizagem e de aquisição do modelo de usabilidade do software pelo usuário final [LEITE '98]. Este problema será agravado pelo fato que, segundo a Engenharia Semiótica [DE SOUZA '01], **a UIL de um software é uma linguagem única** que se refere exclusivamente aos elementos do modelo semântico único do software em questão. Por exemplo, na realização de uma mesma tarefa em dois softwares diferentes que atuam sobre o mesmo domínio, o usuário normalmente empregará uma seqüência de interação diferente que é característica da *UIL* única daquele software. Isto implica que não é possível transferir todo o conhecimento do funcionamento de um software para outro.

A função referencial (ou denotativa) é a função da linguagem em que o enfoque é o contexto da comunicação e, portanto, ela enfatiza o referente informando sobre o objeto do discurso. No processo de interação com o software, o usuário estará veiculando e manipulando informação nas mais variadas formas. Para tanto, ele se envolverá em diálogos com o software que fazem referência a, no mínimo, dois contextos diferentes: 1) o domínio de aplicação do software e 2) o ambiente computacional em que ele executa. Assim, a função referencial exercerá papel primordial para esclarecer a quais objetos o usuário e o software estão se referindo em um determinado momento e em que contexto isto acontece. A existência de contextos diferentes de atuação impõe outra necessidade ao software, a de ter uma *UEL* que seja sensível a esta diferença. A maioria das *UELS* atuais são, no máximo, sensíveis aos elementos léxicos da *UIL* dando respostas idênticas a situações de uso bastante diferentes. Esta não diferenciação no diálogo metalingüístico tem uma grande influência sobre a aquisição do modelo de usabilidade da aplicação pelo usuário, uma vez que ele não terá como distinguir a natureza de suas dúvidas através da diferenciação das situações em que elas ocorrem.

Assim sendo, a despeito de estarmos analisando o uso normal do software extensível, é importante observar que somente quando um usuário compreender suficientemente o modelo de usabilidade do software para saber que a funcionalidade por ele requerida não se encontra disponível é que ele poderá ter a intenção de criar uma extensão a este software. Isto torna as funções metalingüística e referencial essenciais à

tarefa de *EUP* desde a fase inicial de aquisição do modelo de usabilidade do software em virtude da existência de no mínimo dois contextos diferentes de uso e mostra também que é essencial a presença no software extensível de um forte mecanismo de explicação para minimizar a perda da função metalingüística pelo usuário final.

3.2 O uso do software extensível na criação de extensões

No segundo caso do uso de software extensível, o software poderá ter suas funcionalidades ampliadas por meio da criação de extensões. Como o designer não mais está presente no diálogo, estas extensões somente poderão ser realizadas pelo usuário. Logo, neste caso haverá dois emissores: o designer do software (que cria a mensagem original) e o usuário como designer (que criará extensões à mensagem original). Este duplo papel do usuário é um dos elementos críticos no software extensível, pois o usuário não detém conhecimento algum do processo de design de software e, na maioria das vezes, não deseja perder seu tempo para adquiri-lo. Desta forma, o designer deverá prover mecanismos internos ao software que auxiliem o usuário na realização desta tarefa.

Para determinarmos os mecanismos de suporte necessários às atividades do usuário como designer é preciso que analisemos o efeito que seu duplo papel exerce sobre o processo comunicativo. Para isto, primeiramente, é necessário caracterizar o que será considerado uma extensão ao software. Como discutido anteriormente, para o usuário, a *UIL* é o software, assim podemos concluir que, independente da forma de realizar as extensões, elas deverão resultar na alteração do código da *UIL*, pois é este o código que o usuário usa para se comunicar com o software.

O fato de a *UIL* ser a forma de comunicação do usuário e também o componente que será alterado nos aponta a necessidade de um mecanismo de apoio ao usuário que introduza um novo contexto de operação do software — um contexto de realização de extensões. Este novo contexto será necessário para distinguir o uso normal da *UIL*, no qual o usuário estará se referindo a elementos do domínio e/ou do ambiente computacional, do seu uso metalingüístico, no qual o usuário estará se referindo a elementos do meta-modelo do software. Uma vez que o usuário precisa referenciar elementos metalingüísticos, também será necessária a introdução de um novo código — uma **linguagem de extensão do software** (*End-User Programming Language*) — que lhe possibilite comunicar as operações metalingüísticas requeridas para a criação de suas extensões, pois os códigos existentes não lhe possibilitam esta tarefa. A Figura 3 mostra a instanciação do Modelo de Comunicação Verbal de Jakobson para este caso.

A introdução destes dois novos componentes no software extensível resolve apenas parcialmente o problema de suporte ao usuário, pois eles somente fornecem os meios de realizar a tarefa mas não descrevem o processo para realizá-la. Para estabelecermos um modelo satisfatório para a tarefa de *EUP*, é necessário estudar os reflexos da alteração do código da *UIL* sobre o seu uso como meio de comunicação entre o usuário e o software .

Do ponto de vista do usuário, o maior reflexo da possibilidade de alteração do código da *UIL* é que, ao ocupar o papel de emissor na criação de extensões ao software, sua competência terá que mudar, visto que o foco de seu discurso passará a ser os elementos do meta-modelo do software. Assim sendo, não bastará mais a ele saber expressar-se bem na *UIL*, ele também terá que saber se expressar de forma produtiva na *EUPL* (*End-User Programming Language*). Esta exigência decorre do fato de que a

realização de extensões exige do usuário que ele produza descrições linguisticamente corretas das ações que ele quer que a extensão realize no software. Isto implica que o usuário terá que saber tanto a sintaxe como a semântica de *EUPL*, que são elementos metalingüísticos em relação à *UIL*. Por outro lado, ele é um designer privilegiado, pois conhece as reais necessidades do seu usuário e, portanto, sabe exatamente qual a mensagem que deve ser veiculada.

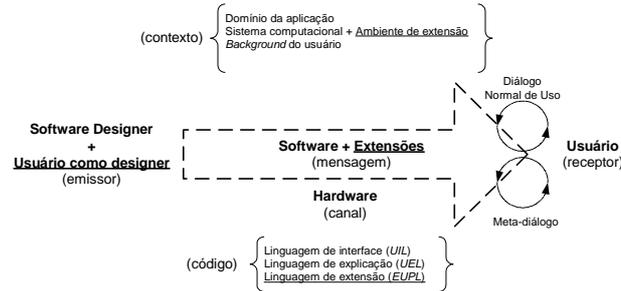


Figura 3: Modelo de Jakobson instanciado para o caso de uso do software extensível na criação de extensões ao próprio software .

Nardi [NARDI '93] nos alerta para o fato de que as dificuldades de programação encontradas pelos usuários finais estão relacionadas à 'baixa expressividade' das linguagens de extensão a eles disponibilizadas. Como solução para este problema, ela propõe que uma *EUPL* seja uma linguagem de propósito especial e fortemente ligada ao domínio da aplicação. Porém, segundo a Engenharia Semiótica, estas linguagens devem não somente ser de propósito específico mas também **linguagens únicas de extensão** [DE SOUZA '01]. Esta unicidade das *EUPLs* está diretamente relacionada à natureza única das *UILs* as quais elas irão estender, e à sua dependência com o domínio da aplicação. Devido a esse aspecto de unicidade, as *EUPLs* apresentam um grau de dificuldade extra para seu aprendizado por parte dos usuários finais.

As necessidades metalingüísticas do usuário, porém, vão mais além. A possibilidade de alteração da *UIL* do software faz com que ele esteja exposto a um processo constante de aprendizado do seu uso, pois, a cada extensão, serão inseridos novos elementos ao código de sua *UIL* possibilitando a ele trocar novas mensagens com a interface do software para a realização de suas tarefas. Este fato faz com que o modelo de usabilidade do software nunca se estabilize e, portanto, o papel da função metalingüística será ainda mais marcante no caso de uso do software extensível do que no anterior. Como reflexo secundário, ele nos mostra que os critérios de avaliação de usabilidade atualmente em uso não se aplicam ao software extensível [DE SOUZA '01], pois pressupõem um modelo semântico estável para o software.

Para o designer, esta possibilidade de alteração acarretará em uma mudança de ponto de vista durante o design do software, pois será necessário alterar a *UEL* não somente para que ela apoie as necessidades metalingüísticas do usuário em relação ao ambiente de extensão e ao uso da *EUPL*, mas também para que ela seja capaz de dar suporte à explicação dos novos elementos inseridos na *UIL* por meio das extensões realizadas pelo usuário. Esta última exigência implica que a ***UEL* terá de ser atualizada a**

cada nova extensão inserida no software. Sem esta atualização não será possível manter a função metalingüística atuando de forma satisfatória, pois ela falhará quando for necessário acessar informações sobre estas extensões.

É interessante observar que os requisitos até agora levantados vão ao encontro direto da proposta de Adler e Winograd [ADLER '92] refletindo o problema da aquisição de conhecimento do software por parte dos usuários finais para a realização de suas tarefas, sejam elas normais ou de extensão. Porém, nenhum deles garante que as extensões ao software produzidas por estes usuários não venham a invalidar o conhecimento já adquirido. A fim de evitar a perda deste conhecimento é indispensável que as extensões somente possam adicionar, reorganizar ou personalizar a *UIL* dentro de uma faixa de tipos possíveis de modificações antecipadas pelo designer, de forma a não corromper seu design original. Devido a esta limitação nos tipos de extensões e ao fato de que estas extensões ocorrerão sobre a *UIL*, a tarefa de realização de extensões em *EUP* será equivalente à **geração de tokens de tipos interativos realizados ou potenciais** que deverão ser intrínsecos à concepção da mensagem original do designer para o usuário [DE SOUZA '01]. Esta visão está de acordo com o mecanismo de programação por demonstração [CYPHER '93], no qual os tipos interativos são usados para inferir extensões pretendidas e somente *tokens* específicos da *UIL* do software, que pertençam a estes tipos, podem ser gerados pelos mecanismos de *EUP*. Estas observações procuram deixar claro o fato de que **extensão é estritamente diferente de programação**, pois na primeira pressupõe-se a existência de algo a ser estendido enquanto na segunda pode-se partir “do nada”. Isto implica que um usuário no papel de designer não poderá introduzir novos elementos que não tenham relação com o modelo de usabilidade original do software.

Contudo, não basta garantir a exclusão de elementos estranhos ao modelo de usabilidade do software para evitar a perda de conhecimento com a criação de extensões, é preciso também que as extensões construídas pelo usuário respeitem o **Ciclo de Interação Mínimo** necessário à criação de diálogos completos na *UIL*. No modelo conceitual aqui proposto, este ciclo mínimo é composto por três passos que podem ser descritos como [DE SOUZA '01]:

software diz alguma coisa para o usuário — isto é, a mensagem do designer, descrevendo as funcionalidades disponíveis, é passada ao usuário através da *UIL*;
usuário diz alguma coisa ao software — isto é, o usuário define a ação que quer executar para realizar parcialmente, ou totalmente, sua tarefa; e
software responde ao usuário — isto é, o software apresenta o resultado da efetivação de sua funcionalidade.

É interessante observar que este ciclo diferencia-se do ciclo de interação dos modelos da tarefa de *EUP* tradicionais devido àqueles, geralmente, considerarem somente os dois últimos passos descritos acima. Desta forma, aqueles modelos não levam em conta o fato de que o designer comunica ao usuário, através da interface, a estrutura de funcionamento do software e que uma falha nesta comunicação resultará na sua baixa usabilidade, como nos mostra Prates *et al.* em [PRATES '00]. Assim, para garantir que o Ciclo de Interação Mínimo seja respeitado, **é necessário que a EUPL tenha uma noção de texto** que seja sintaticamente distinta, ou uma construção de mais alta ordem que aquela equivalente às instruções, bloco ou programa executável, de tal forma que se possa associar uma **interpretação pragmaticamente válida** (na *UIL*) às construções de texto (na *EUPL*).

4. O perfil dos códigos no software extensível

A introdução da *EUP* no conjunto de códigos manipuláveis pelo usuário (possibilitando a alteração da *UIL*) e a necessidade de alteração dinâmica da *UEL* (para suprir as necessidades metalingüísticas do usuário) demonstram que o inter-relacionamento entre os códigos existentes em um software extensível é bem mais complexo que o encontrado em um software não-extensível. Para regular este relacionamento, a Engenharia Semiótica [DE SOUZA '01] define os princípios da Abstração Interpretativa e do Contínuo semiótico, que serão detalhados a seguir.

4.1 O princípio da abstração interpretativa

Na seção anterior assinalamos o fato de que para o usuário final a *UIL* é o software. No entanto, é essencial observar que este fato somente será verdade caso a *UIL* realmente funcione como uma camada de abstração sobre o nível de implementação do software. Isto somente ocorrerá caso o usuário consiga interpretar os códigos empregados na *UIL* recorrendo apenas ao conhecimento disponível no seu contexto de comunicação. Qualquer informação a mais, necessária à interpretação de algum signo da *UIL*, implica que o usuário precisará de conhecimentos que remontam ao nível de implementação do software e, portanto, a barreira de abstração gerada pela *UIL* será quebrada.

Para melhor compreendermos a afirmação acima apresentamos, na Figura 4, um exemplo típico de uma mensagem gerada por um editor de texto comercial em resposta a uma ação do usuário no caso em que o usuário esqueceu de inserir o disquete no *drive* antes de realizar a ação. No diálogo apresentado nesta figura é empregada uma mensagem (*Error 70*) que não faz parte do domínio da aplicação, não consta de parte alguma da *UEL* do software e muito menos faz parte do *background* computacional de um usuário final típico. Este mensagem somente poderá ser interpretada corretamente após uma série de inferências abduativas por parte dos usuários finais levando-os a identificar a ausência do disquete no *drive*.



Figura 4: Exemplo de uma mensagem de erro que emprega códigos que pertencem ao nível de implementação do software.

Como podemos ver, este é um problema que poderá ocorrer em qualquer tipo de software e está diretamente ligado à forma de relacionamento entre os códigos interativos de sua interface e os empregados na sua implementação, que definem a semântica operacional da *UIL*. O **princípio da Abstração Interpretativa**, proposto pela Engenharia Semiótica [DE SOUZA '01], procura avaliar quão bem um código artificial abstrai outro sobre o qual ele está implementado, do ponto de vista de sua interpretação. No caso do software extensível, ele procura avaliar, do ponto de vista de interpretação, quão bem a *UIL* abstrai a faixa de funcionalidades predefinidas e estendidas do software. Deste modo, este princípio serve como balizador da qualidade dos códigos empregados no software (seja ele extensível ou não).

Deste modo, dizer que uma linguagem computacional é uma Abstração Interpretativa de uma ou mais outras linguagens quer dizer que um usuário deve ser capaz de compreender e interagir com a primeira, sem conhecimento algum das especificações expressas nas demais linguagens. No caso de software extensível típico, a *UIL* é composta de uma parte fixa e uma parte extensível (*UILx*). Assim, a parte extensível da *UIL* é uma abstração interpretativa de uma ou mais linguagens computacionais subjacentes — por exemplo, a *EUP*, as linguagens de programação usadas pelos designers de software, ou mesmo as linguagens *assembly* ou de máquina — se um usuário consegue entender completamente todos os signos da *UIL* devido:

1. Aos padrões de signos e combinações de signos que ele encontra enquanto interage com a aplicação;
2. Às explicações disponíveis sobre a *UIL* — da forma como estão disponibilizadas na *EUP* por meio de *help online* e tutoriais, *screen tips*, e documentação em geral da aplicação; e
3. À sua própria experiência com computadores — trazendo à tona o conhecimento associado usado nas metáforas da interface, conhecimento do domínio da aplicação e puro senso comum.

O princípio da Abstração Interpretativa é uma forma qualificada do princípio de abstração conhecido dos teóricos da informática. Trata-se primariamente da **interpretação do discurso situado produzido por sistemas computacionais** (à medida que eles mostram aos usuários alguns resultados da aplicação) e pelos próprios usuários (à medida que eles dão entrada em comandos ou dados para o sistema). O termo qualificador 'interpretativa' é desta forma adicionado ao princípio já conhecido para fins de ênfase e clareza. Em um ambiente de *EUP*, todas as linguagens computacionais contribuem para as tarefas de design. Estas tarefas são dirigidas por intenções, e todas as intenções são determinadas não somente pela experiência dos usuários com o meio e o domínio de atividades — itens 1 e 3 acima —, mas também pela expressividade do código no qual esta experiência pode ser expressa — itens 1 e 2 acima. Assim, de fato, não é somente uma questão de abstrair as camadas semânticas subjacentes aos construtos sintáticos, mas também, e talvez ainda mais importante, de abstrair os objetivos e efeitos que podem ser alcançados tão logo os signos sejam combinados. Logo, este princípio deve valer para as interfaces que tenham boa usabilidade em geral, e não somente para os ambientes de *EUP*.

4.2 O princípio do contínuo semiótico

No modelo de uso de software extensível, descrito na Seção 3 deste artigo, assinalamos a necessidade de controlar o tipo do texto (extensão) que um usuário final pode gerar por meio da *EUP*. Este controle é necessário para poder identificar os textos da *EUP* que são válidos na *UIL*, isto é, os textos que são pragmaticamente adequados, e para garantir o Ciclo de Interação Mínimo do software. Ele também indica a existência de um forte relacionamento entre a *EUP* e a *UIL* no software extensível. Tal relacionamento requer que os elementos da *UILx* possam ser expressos na *EUP* (para que o usuário possa manipulá-los) e que os elementos expressos na *EUP* tenham um reflexo na *UIL* (para que constituam uma extensão). O **princípio do Contínuo Semiótico**, proposto pela Engenharia Semiótica [DE SOUZA '01], procura avaliar o quanto dois códigos artificiais apresentam obstáculos à sua tradução dirigida. Desta forma, ele avalia os obstáculos para traduzir

extensões criadas na *EUPL* em construções funcionais e usáveis da *UIL*. Deste modo, este princípio serve como balizador da qualidade dos códigos empregados no software extensível.

Assim, duas linguagens computacionais são semioticamente contínuas [DE SOUZA '01] se seu acoplamento pragmático for sempre preservado ao se traduzir um texto de uma linguagem para outra em uma direção. No caso de aplicações extensíveis, a parte extensível da *UIL* (*UILx*) é Semioticamente Contínua com a *EUPL* se:

1. Os usuários em geral conseguem extrair sentido dos signos da *UILx* e interagir com a *UIL* sem qualquer conhecimento da *EUPL* ou de sua existência;
2. Ao estender aplicações, os usuários são capazes de compreender os signos da *EUPL* e gerar textos na *EUPL* sem qualquer conhecimento de outras linguagens de mais baixo nível, tais como linguagens de programação, as *APIs* de sistema computacional, etc.;
3. Existe um constituinte sintático para um texto pragmaticamente válido na *EUPL*; e
4. Qualquer usuário que conheça a *UILx* e a *EUPL* sempre pode traduzir uma instância arbitrária de um texto da *EUPL* para uma combinação válida, realizada ou potencial, de signos da *UILx*.

Podemos ver que o princípio do Contínuo Semiótico procura atingir a exclusão das linguagens candidatas a *EUPL* que não distinguem combinações de signos pragmaticamente inadequadas das que são pragmaticamente adequadas. Este acoplamento pragmático entre a *EUPL* e a *UIL* é precisamente a característica que garante a correspondência entre textos bem-formados na *EUPL* e textos na *UIL*.

Diferentemente do princípio de Abstração Interpretativa que focaliza as linguagens de mais alto nível, este segundo princípio focaliza as linguagens de mais baixo nível. O ponto mais importante na definição acima é que se alguém conhece ambos os códigos, ele ou ela sempre pode traduzir qualquer texto arbitrário da linguagem de mais baixo nível para um signo realizado ou potencial da linguagem de mais alto nível. Isto somente pode ser obtido devido a um componente pragmático que é projetado sobre estruturas sintáticas — o <texto> constituinte. Ele pode separar signos que expressam intenções dos que não o fazem e, desta forma, resolver textos em linguagens de mais baixo nível que devem ter uma tradução na linguagem de alto nível dos que não devem.

É importante notar que para que o código da *EUPL* obedeça completamente o princípio do Contínuo Semiótico, é necessário que ela contenha estruturas sintáticas que garantam o acoplamento pragmático entre ela e a *UIL* de forma a refletir o Ciclo de Interação Mínimo anteriormente discutido. O padrão sintático geral da estrutura de um texto em uma *EUPL* Semioticamente Contínua é apresentado na Figura 5.

```

<texto na EUPL>i,j ::= <mensagem anterior do sistema>i,
<disparo da ação>i,j,
<mensagem subsequente do sistema>i,j
<disparo da ação>i,j ::= <mensagem de entrada do usuário>i,
<função ativada>i,j

```

Figura 5: Gramática de alto nível de um texto na *EUPL*.

A relação de continuidade semiótica entre as linguagens presentes em um software extensível pode ser resumida no diagrama apresentado na Figura 6. Nele podemos ver que a *UEL* deverá ser semioticamente contínua tanto com a *UIL* quanto com a *EUPL*. Esta dupla relação deve existir para que se possa satisfazer os critérios de usabilidade definidos por Adler e Winograd.

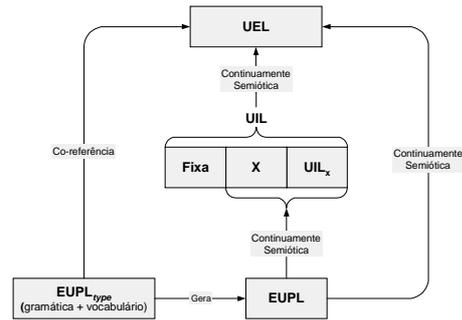


Figura 6: Descrição da relação de Continuidade Semiótica entre a *UEL*, *UIL* e a *EUPL* de um software extensível.

5. Discussão final

Neste artigo, apresentamos um modelo conceitual para a programação por usuários finais. Baseando-nos na abordagem da Engenharia Semiótica que diz que o software é uma mensagem unidirecional e única do designer para o usuário, propomos que o uso do software pode ser visto como um processo de comunicação e representação. Então, apoiados no Modelo de Comunicação Verbal de Jakobson identificamos os elementos envolvidos neste processo e descrevemos a influência de cada um sobre a tarefa de criação de extensões ao software.

Por meio de nosso modelo, mostramos a necessidade de um forte mecanismo de explicação (a *UEL*) no software extensível. Este mecanismo tem duas funções básicas: 1) compensar a ausência do designer no diálogo metalingüístico, auxiliando o usuário na tarefa de aprendizagem do modelo semântico da aplicação e, 2) auxiliar o usuário na aquisição da sintaxe e semântica da *EUPL*. Nosso modelo ainda mostra que este mecanismo deve ser tanto sensível ao contexto, para gerar explicações situadas, quanto evolutivo, para apoiar a explicação das extensões criadas pelo usuário.

Também mostramos que é necessário haver um controle sobre o tipo de código empregado nas *EUPLs*, de modo que as extensões criadas pelo usuário não destruam a mensagem original do designer. Para isto, definimos os princípios da Abstração Interpretativa e do Contínuo Semiótico, que procuram garantir o acoplamento pragmático entre a *EUPL* e a *UIL* e, assim, assegurar a qualidade das extensões geradas.

É importante salientar que este modelo foi desenvolvido visando sistemas monousuário e adaptativos (sistemas em que o usuário indica a intenção de criar a extensão). Outra ressalva importante é que ele não abarca os sistemas que operam com interfaces por manipulação direta.

Por último, é interessante observar que as arquiteturas atuais de software não apóiam as necessidades de explicação existentes para o software extensível levantadas em nosso modelo. Elas tampouco apresentam um processo para guiar a tarefa de criação de extensões. Além disso, as *EUPLs* atuais não incorporam mecanismos para garantir a manutenção dos princípios da Abstração Interpretativa e do Contínuo Semiótico e do Ciclo de Interação Mínimo das aplicação. O modelo conceitual aqui proposto e o tratamento dos aspectos acima citados compõem o Modelo Semiótico para programação por usuários finais descrito em [DA SILVA '01], e serão tema de artigos futuros.

Referências

- Adler, P. and Winograd, T. (1992). "The Usability Challenge", In Adler, P. and Winograd, T. Usability: Turning Technologies into Tools. Oxford University Press. New York, NY. p.3-14.
- Cypher, A. (1993). Watch What I Do: Programming by Demonstration. The MIT Press. Cambridge MA.
- da Silva, S.R.P. (2001). Um Modelo Semiótico para Programação por Usuários Finais. Tese de Doutorado. Departamento de Informática. PUC-Rio. Rio de Janeiro. Maio, 2001.
- de Souza, C.S.; Prates, R.O. and Carey, T. (2000). "Missing and declining affordances: are these appropriate Concepts?", In 3th Workshop on Human Factors in Computer Systems: Many Faces in Interfaces. Gramado – RS, Brazil, October. p.102-111.
- de Souza, C.S.; Barbosa, S.D.J and da Silva, S.R.P. (2001). "Semiotic Engineering Principles for Evaluating End-user Programming Environments", In Interacting with Computers. Vol. 454 (4). p.467-495.
- de Souza, C.S. (1993). "The Semiotic Engineering of User Interface Languages", International Journal of Man-Machine Studies. No. 39, p.753-773.
- de Souza, C.S. (1996). "The Semiotic Engineering of Concreteness and Abstractness: From User Interface Languages to End-User Programming Languages", In Andersen, P.; Nadin, M.; Nake, F. (eds.) Informatics and Semiotics. Dagstuhl Seminar Report No. 135. Schloss Dagstuhl. Germany.
- Eco, U. (1976). Theory of Semiotics. Indiana University Press. Bloomington.
- Fischer, G. (1998) "Beyond 'Couch Potatoes': From Consumers to Designers", In Proceedings of The 5th Asia Pacific Computer Human Interaction Conference. IEEE Computer Society. p.2-9.
- Goodell, H.; Maulsby, D.; Kuhn, S. and Traynor, C. (1999). Report of the CHI'99 Workshop on "End-User Programming and Informal Programming".
- Jakobson, R. (1960). "Closing Statements: Linguistics and Poetics", In Sebeok, T. (ed.) Linguistics and Communication. MIT Press, New York, NY.
- Leite, J.C. (1998). Modelos e Formalismos para a Engenharia Semiótica de Interfaces de Usuário. Tese de Doutorado. Departamento de Informática. PUC-Rio. Rio de Janeiro. Outubro, 1998.
- Lieberman, H. (2001). Your Wish is My Command: Programming by Example. Morgan Kaufmann.
- Nardi, B. (1993). A Small Matter of Programming. MIT Press, Cambridge, MA.
- Peirce, C.S. (1931). "Collected papers". Cambridge, Ma. Harvard University Press. (excerpted in Buchler, J., ed., Philosophical Writings of Peirce. New York: Dover, 1955)
- Prates, R.O.; de Souza, C.S.; Barbosa, S.D.J. (2000). "A Method for Evaluating the Communicability of User Interfaces", In ACM Interactions. (January-February), p.31-38.

Uso de Interfaces Abundantes em Informação para Exploração Visual de Dados

Manoel Gomes de Mendonça Neto, Márcio Oliveira Almeida

Universidade Salvador – NUPERC
Av. Cardeal da Silva, 747 – Federação
Salvador, Ba, Brazil, 20220-141
Tel: (0**71) 339-9543 ou 339-9544
mgmn@unifacs.br

Resumo. O presente artigo discute conceitos de exploração visual de dados, analisando aspectos pertinentes da transformação de dados em atributos visuais que possam ser facilmente interpretados por seres humanos. Um protótipo de uma ferramenta de exploração visual de dados é apresentado e sua abordagem de transformação de dados em informação visual é discutida em detalhe.

1. Introdução

O progresso na tecnologia dos Sistemas de Informação tem proporcionado meios cada vez mais rápidos e baratos de coleta e armazenamento de dados. Dessa forma, empresas e instituições que fazem uso intensivo de computadores tem acumulado uma enorme quantidade de dados. No entanto, esse crescimento no volume dados não tem provocado um aumento correspondente na disponibilidade de informação útil nessas organizações.

A posse de dados não se traduz imediatamente em informação útil. Para isso são necessárias técnicas e ferramentas para exploração eficiente dos dados disponíveis. No momento atual existe uma crescente demanda por tais técnicas. Estas técnicas devem não apenas auxiliar usuários a obter novas informações, mas também auxiliá-los a reconhecer estas informações como úteis.

No bojo do processo descrito acima está a interação entre o ser humano e a máquina. Entre os meios que podem ser usados por seres humanos para explorar dados, a visualização é talvez o mais natural. Meios visuais são rotineiramente usados por seres humanos para interpretar o mundo a sua volta. Como os computadores estão cada vez mais baratos e poderosos, há imensas oportunidades de utilização de computadores na comunicação visual de informações. Shneiderman argumenta que esta utilização se estende a representação visual de dados abstratos [12]. Usando princípios propostos por Tufte [13], tais representações permitem apresentar centenas, milhares, ou até milhões de registros de dados em uma única cena visual. Isto cria um meio eficiente de interpretação de volumes de dados.

Este artigo descreve uma ferramenta para exploração visual de dados. Esta ferramenta é baseada no conceito de *Starfields Displays* e *Dynamic Queries* originalmente propostos por Ahlberg e Shneiderman [1][12]. Esta ferramenta, denominada de Interface de Consulta Visual (em inglês, VQI) é usada como interface básica de busca e análise de pacotes de informação no Sistema de Gestão de Experiências (em inglês, EMS). Um

sistema que visa auxiliar a implementação de Fábricas de Experiências conforme proposto por Basili [2],[3],[4]. A versão do VQI apresentada aqui é parte da segunda geração do EMS, e está sendo usada para gestão de experiências no Centro Fraunhofer para Engenharia Experimental de Software em Maryland, EUA.

2. Transformando Dados em Formas Visuais

Pode-se pensar em visualização como sendo o processo de mapeamento de dados para a forma visual. A Figura 1 mostra graficamente este mapeamento (Card et al. [5]). A utilização deste modelo nos permite simplificar a discussão sobre sistemas de visualização. Na figura 1, as setas que partem da esquerda para a direita representam as etapas de transformação de dados brutos e cenas visuais para interpretação humana. E, as setas que partem da direita para a esquerda representam ajustes destas transformações pelo ser humano. Transformação de Dados mapeia os Dados Brutos em um formato adequado e particular as ferramentas que serão usadas para construir estruturas visuais, resultando em Dados Tabulados.

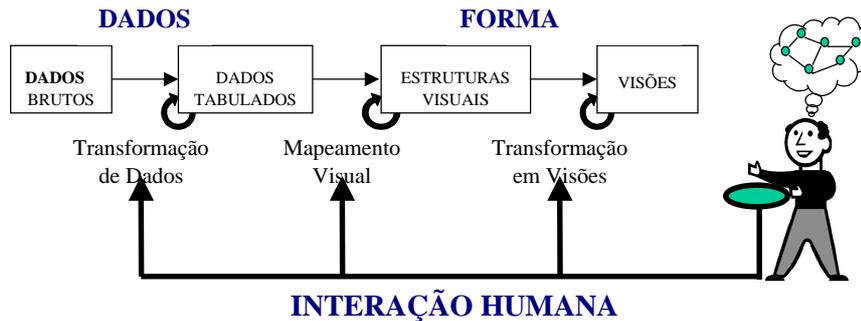


Figura 1. Processo de mapeamento visual de dados.

O Mapeamento Visual transforma os Dados Tabulados em Estruturas Visuais que combinam propriedades gráficas, pontos de referência na tela, e organização espacial destes dados. Finalmente, as Transformações Visuais criam Visões das estruturas montadas permitindo especificação de parâmetros gráficos como posição e escala da visualização. O Ser Humano deve interagir com parâmetros de controle para mudar a natureza das transformações e do mapeamento visual, ou restringir as visões de acordo com o seu interesse.

2.1 Atributos Visuais

Para se mostrar uma informação é necessária preocupação de como a informação será exibida. Por exemplo, para que as pessoas possam rapidamente localizar pontos em um mapa, estes têm que estar representados de uma forma diferenciada. As imagens presentes na Figura 2 mostram algumas maneiras de se representar uma informação de modo que ela se destaque de outras.

Segundo Ware [14], existem várias classificações para os tipos de representações, podemos dividi-las por exemplo em: forma (ex.: largura, tamanho, curvatura, orientação),

cor (ex.: tonalidade, intensidade), movimento (ex.: piscar, direção do movimento), posição espacial (ex.: côncavo/convexo, 2D, 3D), etc.

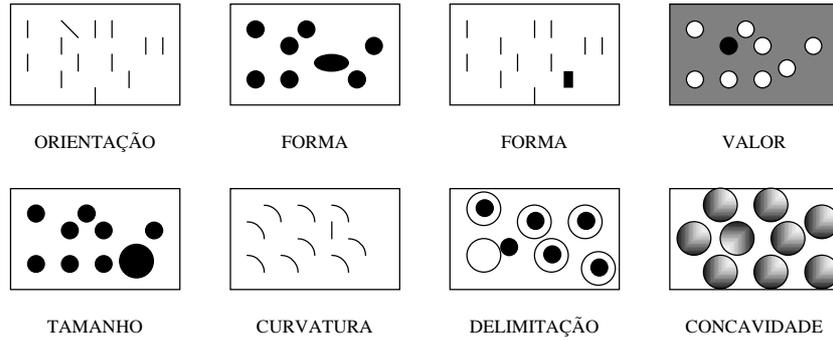


Figura 2. Atributos visuais.

Existem representações, como mostrado na Figura 3, onde a distinção fica mais difícil. Como a busca pela informação é mais demorada nestes casos, estas representações não são desejáveis [14].

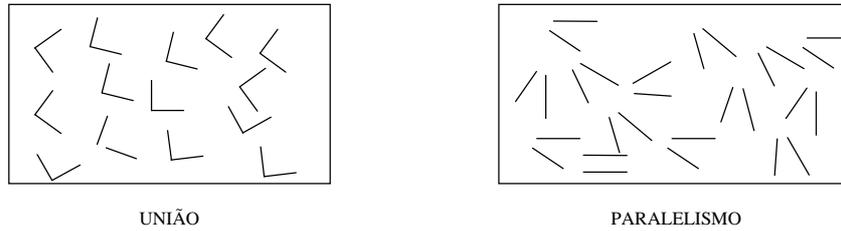


Figura 3. Atributos visuais de difícil visualização.

A escolha de um conjunto de representações e sua estruturação não é uma decisão fácil. A escolha das cores, por exemplo, é uma questão intuitiva e cultural, que requer sensibilidade, e é dependente do domínio de aplicação em que a classificação deve ser usada. Este processo de escolha é mais uma arte do que propriamente uma ciência.

X	0	0,259	0,5	0,707	0,866	0,966	1	0,966	0,866	0,707
Y	12	13	14	15	16	17	18	19	20	21
X	0,5	0,259	0	-0,259	-0,5	-0,707	-0,866	-0,966	-1	-0,966
Y	22	23	24	25	26	27	28	29	30	31
X	-0,866	-0,707	-0,5	-0,259	0	0,259	0,5	0,707	0,866	0,966
Y	32	33	34	35	36	37	38	39	40	41

Quadro 1 – Dados apresentados na forma tabular

3. A Exploração Visual de Dados

Os seres humanos têm uma maior facilidade de processar dados visuais do que textos ou tabelas. Por isso cenas visuais são um importante meio de apresentação dos dados. Usando métodos adequados de visualização, seres humanos podem extrair em poucos milissegundos informações importantes de conjuntos de dados complexos. Boas técnicas de visualização auxiliam o cérebro humano a processar informações complexas de forma rápida.

Seres humanos são muito bons em processar cenas visuais e muito ruins em processar dados tabulares, isso acontece por que sua memória de curto prazo é pequena, para ser exato no ser humano comum esta memória é de apenas 7 ± 2 itens [9], isto implica que após ler apenas alguns números as pessoas começam a esquecer os primeiros. Para exemplificar este conceito, o Quadro 1 apresenta um padrão em formato tabular. Para maioria das pessoas este padrão é de difícil detecção. A Figura 4 mostra o mesmo padrão em um formato gráfico. Agora este padrão pode agora ser facilmente interpretado e mapeado para um modelo senoidal por qualquer pessoa.

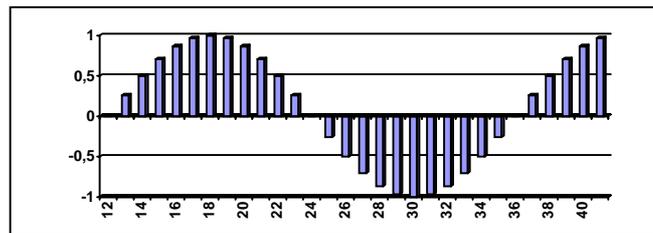


Figura 4. Dados representados na forma gráfica

É importante notar, todavia, que padrões simples como o mostrado na Figura 4 não são comuns. Padrões de interesse são geralmente complexos e variam em diversas dimensões. Ferramentas modernas de visualização devem combinar a capacidade de construir cenas visuais complexas com controles de seleção iterativa dos dados mostrados. Estas funcionalidades combinadas permitem que um ser humano possa explorar interativamente os dados de uma forma eficiente. Nestas ferramentas a exploração pode ser feita tão facilmente que as próprias pessoas conseguem detectar novos padrões de interesse, sem algoritmos específicos para isso. Este tipo de exploração interativa de dados é algumas vezes chamada de mineração visual de dados. Uma boa ferramenta de exploração visual de dados tem as seguintes funcionalidades:

- Permite o uso de diversos atributos visuais (forma, cor, posicionamento, tamanho, etc.) para produzir gráficos multidimensionais facilmente interpretáveis.
- Permite a navegação interativa na tela visual, permitindo aproximação, rotação, reposicionamento, e varreduras sobre a área mostrada.
- Permite um controle interativo dos formatos de apresentação e atributos visuais dos gráficos mostrados.

- Permite o controle interativo dos dados apresentados, habilitando as pessoas a olharem os dados de uma perspectiva geral ou rapidamente mergulhar nos detalhes de um subconjunto de dados.

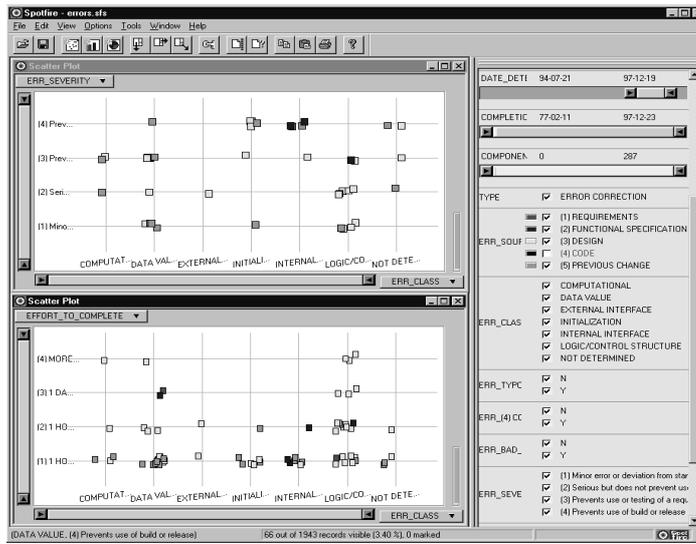


Figura 5. Uma Interface de Exploração Visual de Dados no Spotfire.

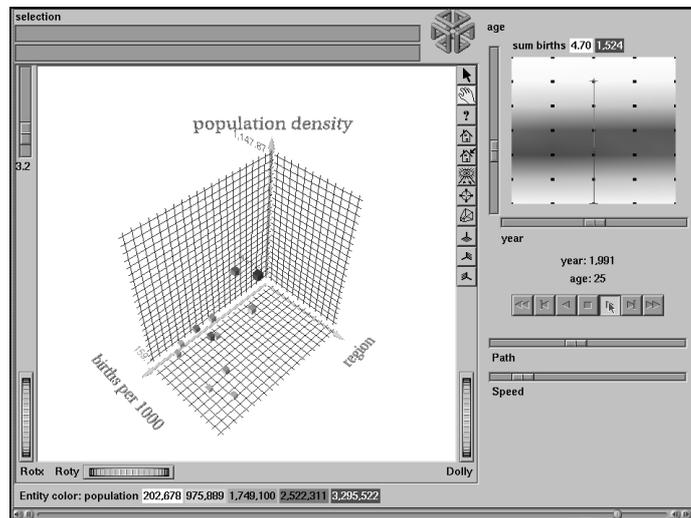


Figura 6. Uma Interface de Exploração Visual de Dados no MineSet da SGI

As Figuras 5 e 6 mostram as interfaces usadas em duas ferramentas comerciais de exploração visual de dados. Essas ferramentas trabalham com o conceito de *Starfields* [1] que utiliza pontos para representação de registros de dados de várias perspectivas. A primeira destas figuras inclui também controles de consulta (veja lado direito da Figura 5) que permitem ao usuário filtrar o conjunto de dados de interesse através de simples seleções com o mouse. Estas consultas gráficas, chamadas de *Dynamic Queries* [11], criam um nível de interação entre usuário e o computador que é comparável ao de um vídeo game, pois a realimentação entre uma seleção no controle de consulta e sua apresentação no *Starfield Display* é praticamente instantânea. Isto cria um ambiente que permite consultas nebulosas e elimina as inconveniências das consultas que retornem respostas vazias [6].

4. Estudo de Caso: Sistema EMS

O EMS (*Experience Manager System*) é um sistema que está sendo desenvolvido em conjunto por pesquisadores do Centro Fraunhofer de Maryland, das Universidades de Maryland em College Park e Baltimore, e a Universidade Salvador. O sistema visa dar suporte as atividades de uma Fábrica de Experiências [2][3][4], e foi desenvolvido originalmente para uma empresa de consultoria em engenharia de software [8][10][15]. No sistema, pacotes de informação estão organizados em categorias que são manipuladas de acordo com a necessidade dos usuários.

Desse sistema faz parte um aplicativo protótipo VQI (*Visual Query Interface*), cujo objetivo é oferecer ao usuário uma forma de consulta visual sofisticada aos dados do sistema. O aplicativo é composto por três módulos: o módulo de controle de consulta, o módulo de controle de atributos visuais, e o módulo de apresentação visual. Estes módulos são responsáveis por criar a interface de apresentação visual dos dados e de fornecer um mecanismo de consulta interativa sobre os mesmos. A arquitetura do aplicativo VQI é mostrada na Figura 7.

De forma similar ao sistema Spotfire mostrado na Figura 6, a VQI do EMS apresenta duas partes principais: a parte onde os dados são representados visualmente e a parte de consulta que permite ao usuário filtrar os dados de acordo com suas necessidades.

O conjunto de **dados** pode ser imaginado como uma tabela, onde cada linha representa um registro e cada coluna representa uma **variável**. Um **controle de consulta** é criado para cada variável, baseado no tipo dos dados e no intervalo de variação de seus valores. A cada momento um sub-conjunto das variáveis disponíveis é mostrado na tela visual. **Controles de atributos visuais** são usados para selecionar quais variáveis serão mostradas na tela visual. A tela visual representa simultaneamente até três variáveis com a utilização do eixo X para representar a primeira variável, o eixo Y a segunda e a cor a terceira. O lado esquerdo das Figuras 8 e 9 mostram exemplos destas telas visuais. Os pontos apresentados nesta tela representam registros de dados de uma determinada classe de informação (ex. documentos, projetos, processos, etc.). Os controles das três variáveis mostradas são feitos através de menus localizados sobre a própria tela visual.

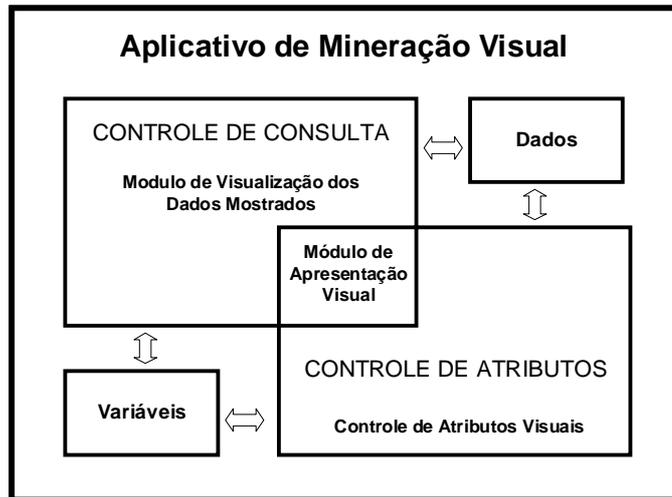


Figura 7. Arquitetura do aplicativo VQI.

A segunda parte da interface contém os controles de consultas para todas as variáveis usadas para classificar uma certa classe de informação sob análise. Estes controles são usados para interativamente filtrar o conjunto de dados apresentado na tela visual da esquerda. Os controles de consulta podem ser ordenados ou ocultos de acordo com a preferência dos usuários e a importância das variáveis para a tarefa em mãos.

4.1 Utilização de Atributos Visuais

No VQI, o plano cartesiano da tela visual é utilizado para representar duas variáveis, e suas cores são usadas para representar uma terceira variável. Além disso, o tamanho e dispersão dos pontos podem ser ajustados para facilitar a visualização dos registros de dados.

4.1.1 Cores para Dados Categóricos

Existem basicamente dois tipos de dados com os quais trabalhamos, são eles os dados categóricos e dados numéricos. Os dados categóricos são dados que não possuem sentido de ordem ou unidade enquanto os dados numéricos possuem. O VQI utiliza esquemas diferentes de cores para cada um deles.

Na representação dos dados categóricos é usado um esquema de cor que utiliza cores claras e bastante distintas. Desta forma as categorias dos dados são facilmente distinguidas umas das outras como mostra a Figura 8.

As cores utilizadas nas representações são realimentadas no controle de consulta equivalente a variável de coloração. Pode-se também escolher que cor será associada a um determinado valor da variável categórica. Isto permite ao usuário associar certos conjuntos de dados as cores que melhor representem para ele a informação sob análise. Por exemplo,

documentos brasileiros poderiam ser representados em verde, americanos em azul, e alemães em vermelho.

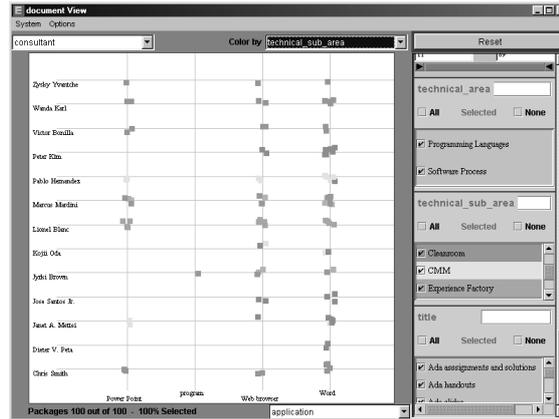


Figura 8 – Representação de dados categóricos.

4.1.2 Cores para Dados Numéricos

Para representar dados numéricos é utilizado um esquema de cores contínuo que parte de uma cor mais clara para uma escura dentro de um mesmo tom. Os dados de valores mais baixos são representados pelas cores mais claras e os de valores mais altos pelas cores mais escuras. Como exemplo, a figura abaixo representa a variável "data" em tons de azul. A parte mais clara representa as datas mais antigas e a mais escura representa as datas mais recentes.

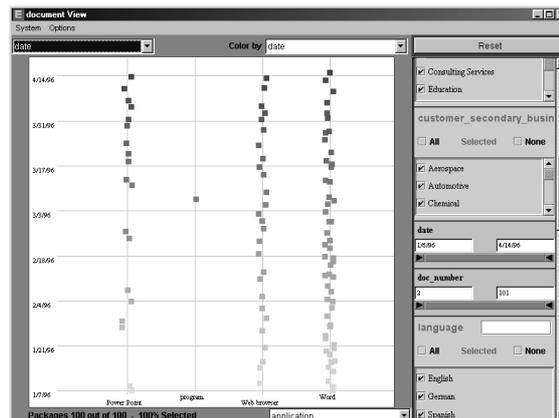


Figura 9 – Representação de dados sequenciais.

4.1.3 Ajuste de Tamanho e Dispersão dos Pontos (*Jitter*)

Para melhor visualizar os pontos, pode-se alterar o seu tamanho como mostrado na Figura 10. Cenas visuais com muitos pontos requerem pontos menores na tela visual.

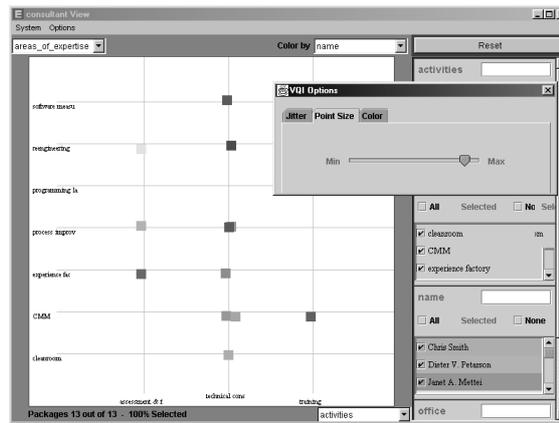


Figura 10. Ajuste de tamanho dos pontos.

Quando existem muitos pontos exibidos, pode haver uma sobreposição de pontos dificultando a manipulação dos mesmos pelo usuário, veja exemplo na Figura 11. Para resolver este problema é utilizado um controle de dispersão de pontos (*jitter*) que aleatoriamente dispersa o ponto da sua posição original, veja exemplo na Figura 12.

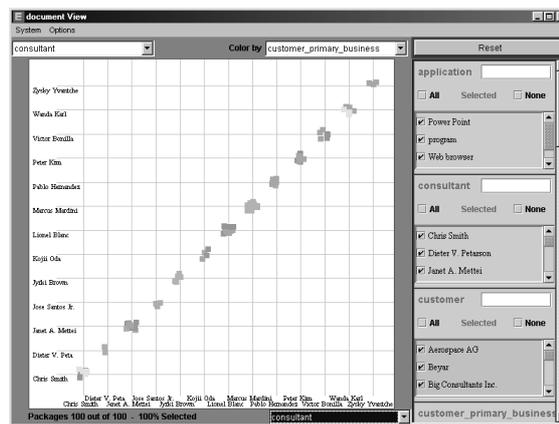


Figura 11. Dados excessivamente sobrepostos.

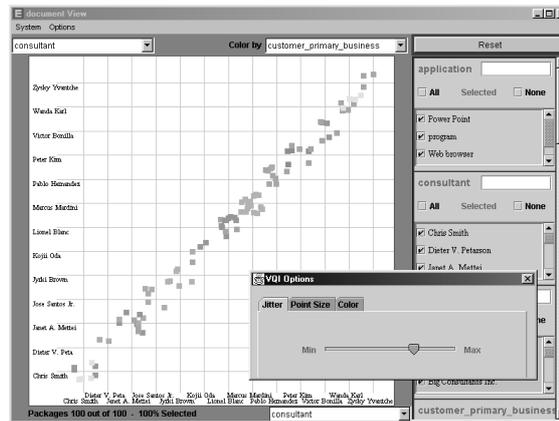


Figura 12. Ajuste de dispersão dos pontos.

4.1.4 Obtendo Detalhes

O sistema possibilita que dados detalhados de um item sejam mostrados ao se pressionar o botão do mouse sobre o ponto que representa este item na tela visual. Pressionando-se o botão direito, é obtido um detalhamento dos dados como mostrado na Figura 13. Pressionando o botão esquerdo, é obtido um formulário descrevendo este item detalhadamente. A partir deste formulário, pode-se recuperar do servidor de informações o corpo dos documentos associado ao registro de dados sendo inspecionado.

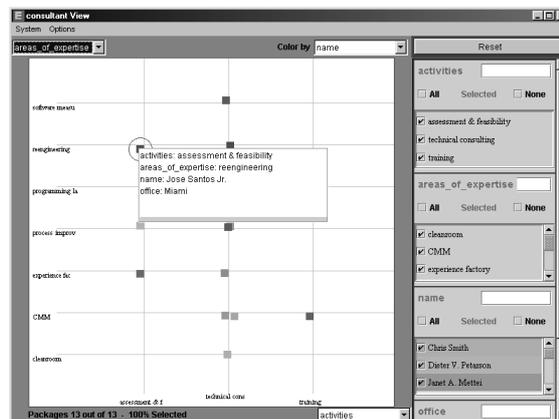


Figura 13. Obtendo detalhes sobre um ponto.

4.2 Controles de Consulta

Os controles de consultas [4] são "widgets" que permitem a execução de operações de seleções sobre conjunto de dados sendo manipulado. Estes controles de consulta são

vistos no lado direito da Figura 13. Eles são formados dinamicamente de acordo com o tipo e número de valores associados a uma determinada variável. Para cada variável associada à classe de informação sendo analisada é criado um controle de consulta. Dados categóricos e numéricos possuem controles diferentes.

4.2.1 Controles de Consulta para Dados Categóricos

Dados categóricos, aqueles que contém diferenças nominais entre os seus itens, usam um controle baseado em caixas de seleção. A Figura 14 mostra o controle de consulta para dados categóricos, neste caso é representada a variável *application*. Ao se clicar em uma caixa de seleção deste controle, os pontos somem ou aparecem da área de visualização, conforme eles satisfizerem ou não o critério estabelecido por este controle de consulta.

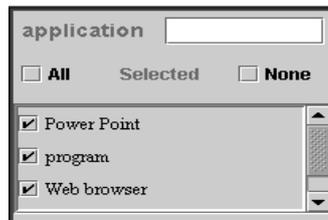


Figura 14. Controle de consulta para dados categóricos.

Este controle possui recursos para busca da caixa desejada no campo ao lado do nome da variável (no nosso caso, *application*). Isto permite a navegação rápida entre as opções do controle de consulta, quando um número grande de caixas existir nele. Outra característica deste tipo de controle é a realimentação do número de opções selecionadas, e das cores usadas para cada opção. Este último é utilizado somente se a variável está sendo usado como atributo de coloração na janela de visualização. Um exemplo disto pode ser visto para a variável *Name* na Figura 13.

4.2.2 Controles de Consulta para Dados Seqüenciais.

Dados numéricos utilizam o controle de consulta mostrado na Figura 15. Este controle usa uma barra de intervalo. Nela se pode selecionar uma faixa de valores para a variável representada. Todos os itens que não atendam as seleções feitas nesta barra também desaparecem da tela. O exemplo abaixo mostra este tipo de controle para a variável *date*.



Figura 15. Controle de consulta para dados numéricos.

Na barra de intervalo, pode-se pressionar a seta da direita para se ajustar o limite superior da seleção, a seta da esquerda para se ajustar o limite inferior da seleção, a barra central para se ajustar os dois limites ao mesmo tempo.

4.3 Validação do Sistema

A versão original do EMS, desenvolvido para uma empresa de consultoria em engenharia de software, foi validada em um experimento controlado em que usuários executaram um conjunto de casos de uso da ferramenta. Neste estudo, os usuários receberam uma série de exercícios representando cenários de trabalho. Eles foram incentivados a usar um protocolo "think aloud" e suas ações foram documentadas [7]. Após cada sessão, eles responderam um questionário qualitativo de avaliação do sistema. Este experimento e seus resultados estão reportados em [10].

Apesar de ser usado para validar o sistema como um todo, este experimento teve forte foco na avaliação da VQI. Em geral os usuários ficaram bastante satisfeitos com a abordagem usada na interface de consulta visual. Todavia, a interface descrita neste artigo é bem mais sofisticada. A versão original não usava cores, possuía controles de consulta bem mais simplificados, e muito menos recursos de configuração dos controles de consulta e da tela visual. Novos estudos para validar a nova interface e seus recursos estão sendo considerados para o futuro.

5. Conclusão

Com o acúmulo de dados no mundo atual, técnicas de exploração visual de dados vêm se tornando cada vez mais importantes. Elas permitem melhor exploração dos dados coletados, criando uma ponte entre dados brutos e informação útil. No momento atual é crucial o desenvolvimento de interfaces de consultas visuais cada vez mais eficientes, interfaces que são ao mesmo tempo abundantes em informação e fáceis de usar, permitindo a usuários e instituições extrair mais e mais informações úteis dos dados que coletam.

Exploração visual de dados representa a forma mais intuitiva de exploração de dados. Todavia, devido às dificuldades de sua implementação, só agora estão aparecendo ferramentas com forte suporte a exploração visual de dados. Mesmo assim, pode-se dizer que esta área ainda está na sua infância. Existe muito de arte na criação destas ferramentas, é preciso que se transforme o processo de construção e avaliação destas ferramentas em ciência.

Este trabalho discutiu uma destas ferramentas mostrando seu uso em um projeto ligado ao conceito de fábricas de software. A interface mostrada aqui ainda está em desenvolvimento e novas funcionalidades devem ser acrescentadas em breve. A principal será a adição de funcionalidade para trabalhar com registros de dados com múltiplos valores para a mesma variável. Estamos também estudando quais padrões de projetos são mais adequados a construção de interfaces para exploração visual de dados.

6. Bibliografia

- [1] Ahlberg, C. e Shneiderman, B., Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In Proceedings ACM CHI'94, pp. 313-317 plus color plates, April 1994.

- [2] Basili, Victor R. Software Development: A Paradigm for the Future. *COMPSAC '89*, Orlando, Florida, pp. 471-485, September 1989.
- [3] Basili, Victor R. The Experience Factory and its Relationship to Other Improvement Paradigms. *4th European Software Engineering Conference (ESEC)*, Garmish-Partenkirchen, Germany. The Proceedings appeared as the Springer-Verlag Lecture Notes in Computer Sciences Series 717, September 1993.
- [4] Basili, Victor R., and Gianluigi Caldiera. Improve Software Quality by Reusing Knowledge and Experience. *Sloan Management Review*, MIT Press, Volume 37, Number 1, Fall 1995.
- [5] Card, Stuart K.; Mackinlay, Jock D.; Shneidermann, B. Readings in Information Visualization : Using Vision to Think. Morgan Kaufmann Publishers, 1999.
- [6] Greene, S., Tanin, E., Plaisant, C., Shneiderman, B., Mushlin, R., Olsen, L., Major, G., and Johns, S. (1997) The End of Zero-Hit Queries: Query Previews for NASA's Global Change Master Directory *International Journal of Digital Libraries*, Vol. 2, No. 2+3 (79-90), 1999, CS-TR-3856, UMIACS-TR-97-84.
- [7] Hackos, J.T. and J.D. Redish, *User and Task Analysis for Interface Design*. New York:John Wiley and Sons, 1998, chapter 9, pp. 258-9.
- [8] Mendonça, Manoel G. , Seaman, Carolyn B., Basili,Victor R., and Kim, Yong-Mi. A *Prototype Experience Management System for a Software Consulting Organization*. To Appear, In Proceedings of the 13th International Conference on Software Engineering and Knowledge Engineering. Buenos Aires, Argentina, June 2001.
- [9] Miller, G.. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 101(2), pp. 343-352, April 1994.
- [10]Seaman, Carolyn B., Mendonça, Manoel G., Basili, Victor R., and Kim, Yong-Mi. An Experience Management System for a Software Consulting Organization Presented at the *Software Engineering Workshop*, NASA/Goddard Software Engineering Laboratory, Greenbelt, MD, December 1999.
- [11]Shneiderman, B. The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology*, 1, p. 237-256, 1982.
- [12]Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software*, Los Alamitos, v. 6, n. 11, p. 70-77, Nov. 1994.
- [13]Tufte, Edward. *The Visual Display of Quantitative Information*. Graphics Press, 1992.
- [14]Ware, Colin. *Information Visualization: Perception for Design*. Academic Press, 2000.
- [15]Webby, R., C. Seaman, M. Mendonça, V.R. Basili, and Y. Kim. Implementing an Internet-Enabled Software Experience Factory: Work in Progress. Position paper at the *2nd Workshop on Software Engineering over the Internet (ICSE'99)*, Los Angeles, CA, May 1999.



*IV Workshop
sobre Fatores Humanos
em Sistemas Computacionais
Interfaces para Todos*

Posters

A norma ISO 9241:10 e a participação do usuário na especificação da usabilidade de sistemas de recuperação da informação (SRI) na Web

Maria Esther Russo Lima¹, Walter de Abreu Cybis²

¹Bibliotecária do Tribunal Regional Eleitoral do RioGrande do Norte

²Univ. Federal de Santa Catarina

Departamento de Informática e Estatística

estherusso@yahoo.com.br, cybis@inf.ufsc.br

Abstract. *The effectiveness of the recovery of information through sites of the web, such as virtual libraries and search sites depends on the quality of the interface of the system. The work presents a foundation of the ergonomics applied to the technology of the information and takes, as reference, the usability patterns for software recommended by ISO 9241-10. As a result of the research, it presents a list of requirements indicated by the users and considered important for conception of an ergonomic interface; through the adoption of a collaborative approach, it is also suggested the use of Participatory Design techniques (PD) for the project development.*

Keyword: *Ergonomics Interface – SRI – Library Virtual – Participatory Design.*

1. Introdução

Diante das novas tecnologias disponíveis para comunicação em rede e processamento de dados, as organizações responsáveis pela socialização da informação devem estar atentas ao novo cenário, de modo a acompanhar o seu desenvolvimento e utilizar estas ferramentas para otimizar seus processos de indexação, armazenagem e disseminação de informações através da Web. Nesta categoria encontram-se as bibliotecas virtuais e sites de busca e recuperação da informação. Com o crescente uso dos Sistemas de Recuperação da Informação (SRI), estas organizações devem projetar seus sistemas para atender as expectativas dos usuários, independente do seu grau de experiência, oferecendo interfaces ergonômicas e eficazes. Twidale e Nichols (1998) consideram que, para resolver os problemas de usabilidade dos SRIs, estes precisam ser projetados para serem utilizados pelos "iniciantes perpétuos".

Com base nesta concepção, este trabalho aborda a questão da análise e especificação de requisitos de usabilidade de SRI, apresentando como possível solução para realização desta fase de projeto a aplicação dos princípios de diálogo recomendados pela norma ISO 9241:10 e sugere que seja adotada a abordagem do *design* participativo (PD) centrada nos usuários para concepção de interfaces ergonômicas. Com estas premissas realizou-se um estudo de caso no CEFET-RN a partir da aplicação de questionário derivado das recomendações da referida norma. Os resultados desta atividade foram tratados para produzir uma especificação geral da usabilidade de um novo SRI, a servir a esta mesma comunidade. A justificativa é de que o envolvimento dos usuários

durante o processo de análise e especificação da usabilidade seja um bom ponto de partida para que se tenha um SRI ergonômico.

2. A ISO 9241 e os Requisitos de Usabilidade para Interfaces de SRI

A parte 10 (princípios de diálogo) da norma ISO 9241 é um padrão ergonômico para a usabilidade de *software*. Neste trabalho, ela é empregada para apoiar o processo de especificação da interface humano-computador do sistema proposto (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 1997 e CYBIS, 2000). Estes princípios foram básicos para elaboração do questionário aplicado no estudo de caso, proporcionando uma estrutura para definir os requisitos de usabilidade da interface, no que se refere a sua capacidade de orientar as ações de busca e recuperação da informação pelos usuários.

Para Oppermann e Reiterer (1997) uma das principais vantagens da atividade de padronização é a garantia para fornecedores e compradores de software de que o produto atende as exigências e aos princípios ergonômicos de qualidade.

3. Requisitos de Qualidade Prioritários para Usuários de SRI

Foi realizado um estudo de caso com trinta e seis usuários da internet, com idade acima de 13 anos e grau de instrução entre nível médio e pós-graduação (mestrado), alunos e professores do CEFET-RN. Utilizou-se como instrumento de pesquisa um questionário estruturado em três partes: identificação do tipo de usuário, sugestões de serviços a serem oferecidos e na terceira parte, os requisitos de diálogo recomendados pela norma 9241:10. Os itens distribuídos em sete grupos foram priorizados pelos usuários de acordo com o grau de importância atribuído a cada um na escala de 1 a 5. O tratamento dos resultados dos questionários permitiu a explicitação de uma lista de requisitos gerais (quadro 1) para a usabilidade do SRI proposto. A coluna um contém os requisitos de acordo com a prioridade e a coluna dois, as recomendações para atender aos mesmos.

QUADRO 1 – REQUISITOS PARA INTERFACES DE SRI	
1. Sistema de Apoio ao Diálogo na Realização da Tarefa	
Recursos para recuperar dados perdidos.	O comando desfazer é um exemplo que pode atender a este requisito.
Possibilidade de escolher o estilo de diálogo que lhe seja mais adequado.	As propriedades e opções disponíveis devem ser apresentadas de modo didático, para que o usuário reconheça sua necessidade e queira utilizá-las.
Diálogos curtos, objetivos e auto-explicativos.	O <i>site</i> do Google do é um exemplo de interface que facilita a escolha do tipo de diálogo desejado pelos usuários (www.google.com.br).
Diálogos conduzidos passo a passo, com instruções detalhadas.	A interface do Google para pesquisa avançada é um bom exemplo de como o SRI deve ser projetado para atender a este recurso, por ser auto explicativo (<i>idem</i>).
2. Descrição Automática do Tipo de Operação	
Mensagem de erro na mesma linguagem do usuário.	A mensagem de erro deve ser clara e compreensiva, principalmente para os iniciantes. Quando estiver diante do impasse o usuário precisa saber o que ocorreu, podendo ele mesmo resolver ou saber informar o que aconteceu.
Recursos para proteção contra erros, levando a perda de dados.	Oferecer <i>feedback</i> automático quando o usuário executar uma ação que pode levar a perda de dados.
Convites para entradas claros e eficientes.	O <i>design</i> dos comandos de entrada devem ser posicionados em local de fácil visualização e adequados para o tipo de operação.
Informações sobre o sistema e instruções claras, objetivas, completas e contextuais.	Estas informações podem vir no sistema de ajuda e sempre em linguagem familiar ao usuário. Deverão ser sensíveis ao contexto, reconhecer onde o usuário está, o que está tentando fazer e oferecer informações consistentes.
Possibilitar a escolha entre explicação breve ou detalhada do erro cometido.	Para resolver este problema o sistema deve oferecer opção de escolha para o usuário, respeitando o seu nível de conhecimento sobre o sistema.

Informar o tempo de processamento dos dados solicitados.	Considerando que a recuperação da informação pode ser demorada, por conter figuras, gráficos, quadros etc., o sistema deve informar o tempo de processamento é imprescindível.
3. Capacidade de Controle das Ações por parte dos Usuários	
Desfazer as ações e poder recuperar última informação deletada.	O comando desfazer utilizado em alguns editores de texto são exemplos que podem ser seguidos para atender a este requisito.
Possibilidade de retornar as interações a qualquer momento.	O comando Alt+Tab utilizado por alguns sistemas é um recurso eficaz para possibilitar o usuário retornar a interação sempre que desejar.
Possibilitar ao usuário escolher a quantidade de registros que quer ver.	As interfaces devem permitir ao usuário controlar quantos documentos devem retornar, quantidade por página, qual a sequência, definir o agrupamento, etc.
Possibilitar ao usuário a escolha do próximo campo que deseja preencher.	Evitar o excesso de automatismo das ações e oferecer opção para que o usuário possa dar o próximo passo de acordo com a necessidade do momento
4. Conformidade com as expectativas dos usuários	
Padronizar o uso de teclas para realizar ações semelhantes.	Adotar padrões comuns a outros sistemas (F1 para ajuda e clicar o botão direito do mouse para obter detalhes) são exemplos de recursos que atendem a este requisito.
Oferecer ajuda para o significado de termos técnicos quando o uso é inevitável.	O sistema deve oferecer recursos para que o usuário obtenha informações sobre os termos que não conhece. A informação sobre a existência deste recurso deve estar visível para o usuário.
5. Tolerância a Erros	
Recursos para proteger contra o acionamento involuntário de comandos perigosos.	Oferecer <i>feedback</i> (mensagem na tela ou aviso sonoro) sempre que alguma ação perigosa for realizada, principalmente, se isto pode levar a perda de dados.
Correção automática de erros comuns na entrada de dados quando o usuário fugir ao padrão adotado.	O recurso é útil para corrigir erros do tipo: entrada pelo primeiro nome do autor, imprecisão de busca por palavra-chave, etc. O sistema deve informar a forma correta de busca de modo a não dar resposta negativa ao pesquisador.
Alertar o usuário quando ele fugir da sequência de ações de busca programada.	A busca programada é um recurso que pode ajudar ao usuário iniciante seguir os mesmos passos durante o processo de busca. O sistema deve alertá-lo quando fugir a esta sequência a fim evitar que o seu objetivo não seja atingido.
Robustez do sistema ao aceitar erros de entrada sem travar.	O sistema deve suportar qualquer tipo de erro sem travar, fazendo com que o usuário não perca os dados recuperados e tenha que reiniciar todo o processo.
6. Adequação a Individualização	
Recursos para atender deficiências visuais.	Recursos como o zoom, avisos sonoros, sistema de reconhecimento de voz para entrada e saída de dados são úteis para deficientes visuais. Para os completamente cegos o sistema deve codificar informação para impressão em <i>braille</i> .
Possibilitar ao usuário programar teclas e funções que facilitem a operação.	Oferecer recurso para que usuário obtenha uma interface personalizada de acordo com suas preferências e permitir salvar o formato definido para uso posterior.
7. Adequação ao Aprendizado	
Opções de nível de pesquisa de acordo com a habilidade do usuário.	Oferecer o recurso de busca simples e avançada com explicações claras de como utilizar e obter melhores resultados.
Fornecer opção de teclas de atalho significativas e compreensíveis para os comandos mais frequentes.	O uso de teclas de atalho do tipo <i>control</i> , ou <i>alt</i> + primeira letra do comando é um exemplo que pode ser utilizado e deve estar bem explícito para o usuário.
Oferecer lições passo a passo para uso do sistema por iniciantes.	Este recurso deve ser aplicado tanto para ajuda ao sistema como para instruir sobre como conduzir a pesquisa e obter os melhores resultados.
Oferecer módulos de exercícios com estratégia de aprendizado.	Disponibilizar este recurso sempre que possível, considerando que sempre terá alguém utilizando o sistema pela primeira ou queira aprender mais.

Fonte: LIMA, 2001

Propõem-se que a futura passagem destes requisitos para a equipe de projeto e validação pelos próprios usuários, seja reforçada com a apresentação de bons e maus exemplos de interfaces de sistemas para recuperar informação disponíveis na *web*.

4. Considerações Finais

A crescente oferta de SRIs na *web* tem colocado em cheque a qualidade de alguns para cumprir seus objetivos de comunicar, informar, disseminar conteúdos e democratizar a informação a leigos e especialistas. Nesta proposta é colocado como essencial que estes sejam planejados e desenvolvidos com base em critérios ergonômicos, de modo a garantir a sua usabilidade por qualquer tipo de usuário. Para a concretização de tais objetivos, recomenda-se que as experiências dos usuários com sistemas semelhantes e suas sugestões de melhoria devam ser utilizadas na definição de requisitos de qualidade do SRI desde a fase de levantamento de necessidades e durante todo o seu ciclo de vida. Espera-se que com estes procedimentos obtenha-se um sistema simples de usar, eficaz nas respostas e proporcione a plena satisfação dos usuários.

Para a concepção de SRIs ergonômicos e com estas qualidades, sugere-se a formação de grupos de trabalho envolvendo os usuários e que a equipe seja composta, além dos analistas de sistemas e programadores, também, por especialistas das áreas de: IHC, ciência da informação, *design* gráfico, psicologia etc. Alguns conceitos sobre PD e técnicas para trabalhar com grupos formados para desenvolver projetos dessa natureza podem ser vistos em Muller, Haslwanger e Dayton (1997), *Information & Design* (2001), Levinger (2000), entre outros.

5. Referências Bibliográficas

- CYBIS, Walter de Abreu et. al. *Ergonomia de interfaces humano-computado*. Disponível em <<http://www.ufsc.led.br>>. Acesso em: 23 abr. 2000.
- INFORMATION & DESIGN. Disponível em: <<http://www.infodesign.com.au/usability/participatorydesign.htm>>. Acesso em: 15 jan. 2001.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Ergonomic requirements for office work with visual display terminals (VDTs)*: parte 10: dialogue principles. ISO 9241. Switzerland, 1997.
- LEVINGER, David. *Participatory Design History*. Disponível em: <http://www.cpsr.org/conferences/pdc98/>. Acesso em: 14 dez. 2000.
- LIMA, Maria Esther Russo. *Análise e especificação de requisitos ergonômicos para sistema de recuperação da informação (SRI) na Web*. 2001. 208 f. Florianópolis : UFSC, 2001. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2001.
- MULLER, Michel J.; HASLWANTER, Jean Hallewell; DAYTON, Tom. Participatory practices in the software lifecycle. In: *Handbook of Human-Computer Interaction*. cap. 11, 1997. Mimeografado.
- OPPERMANN, Reinhard, REITERER, Harald. Software evaluation using the 9241 evaluator. *Behaviour & Information Technology*. [s.l.], v. 16, n. 4/5, p. 232-245. 1997. Mimeo.
- TWILDALE, Michael; NICHOLS, David. Design interfaces to support collaboration in information retrieval. *Interacting with Computers*. [s.l.], v. 10, p. 177-193, 1998.

Alternativas de Interação numa Sociedade que Envelhece

André Junqueira Xavier¹, André Raabe², Márcia Barros de Sales³

¹Pós-Graduação de Ciência da Computação, ³Pós-Graduação de Engenharia de Produção -
Universidade Federal de Santa Catarina – UFSC
CTC - Campus Universitário – Trindade - 88040-900 – Florianópolis – SC - Brasil

²Ciência da Computação – CTTMar – Universidade do Vale do Itajaí
andrejx@inf.ufsc.br, araabe@inf.univali.br, marciab@eps.ufsc.br

***Abstract.** This paper presents the search for interaction alternatives for elderly people. The main objective is to develop interfaces considering elderly abilities. Informative and consultative stages of the participatory design are also presented.*

***Resumo.** Este trabalho apresenta a busca de alternativas de interação para usuários idosos. O principal objetivo é desenvolver interfaces que valorizem as habilidades do idoso. O trabalho foi desenvolvido baseado em um design participatório, onde são apresentadas as etapas informativa, consultiva.*

1. Introdução

Por que aproximar idosos da informática? Para minimizar o processo de exclusão que este grupo etário está sofrendo. Para diminuir perdas comuns nesta faixa etária por meio do acesso às informações. As novas tecnologias de informação e comunicação podem ser instrumentos de distanciamento ou de construção de pontes entre as gerações.

Existem hoje no Brasil aproximadamente 14,5 milhões de pessoas com idade superior a 60 anos [IBGE 2001]. Neste grupo há uma parcela significativa de pessoas em fase de pré-aposentadoria ou já aposentadas. Estes dados comprovam a importância de se buscar alternativas de acessibilidade ao computador para esta população. Segundo Nielsen (2000), 50% das pessoas com idade acima de 65 anos apresentam algum tipo de incapacidade funcional, dificultando a sua interação com o computador.

Baseado na experiência obtida na vivência com os idosos em oficinas de introdução a informática, identificou-se que alguns idosos apresentam dificuldades em acompanhar o processo de evolução tecnológica. Esta dificuldade pode ser ocasionada por alguma espécie de incapacidade funcional ou por resistência a tecnologia. Presume-se que isso pode proporcionar um afastamento do idoso do meio produtivo, muitas vezes antecipando ou motivando-o a aposentadoria precoce.

Porém, há funções que permanecem intactas ou que sofrem mínimo comprometimento até idades avançadas, são elas: vocabulário e fluência verbal, que pode inclusive aumentar em relação aos jovens, e a capacidade de escrita e leitura [Mattos 1999].

A partir dessas informações, elaborou-se a hipótese de que o uso de comandos de voz seria uma alternativa viável de interação, pois possibilita valorizar a habilidade de fluência verbal identificada nos idosos. Para isso, estruturou-se uma oficina, valendo-se da

experiência de uma equipe interdisciplinar, para avaliação do potencial da interação do idoso com o computador por meio de comandos de voz.

2. Técnica utilizada na oficina

Este trabalho baseou-se na técnica de design centrado no usuário, contemplando duas etapas de envolvimento com este:

Etapa Informativa: a qual baseou-se na literatura de gerontologia e geriatria, e em levantamentos estatísticos sobre a realidade do idoso no Brasil;

Etapa Consultiva: onde foi realizada uma oficina dividida em dois momentos.

1. Demonstração do funcionamento de um software que interpreta de comandos de voz (IBM ViaVoice), e coleta de depoimentos dos idosos, obtidos por intermédio de questionários, sobre a expectativa de interagir com o computador por meio de comandos de voz.
2. Interação dos idosos com o computador por meio de comandos de voz em um exercício de composição de textos. Participaram do experimento quatro idosos. A interação foi filmada conforme ilustra a figura 1.



Figura 1. Processo de interação com o computador

3. Resultados Preliminares e Perspectivas Futuras

Os depoimentos coletados demonstraram que existe uma tendência de que a interação por meio de comandos de voz seja mais agradável ao idoso do que o uso do teclado e mouse. Esta característica possibilita reduzir a resistência que alguns idosos apresentam ao processo de interação com o computador, além de favorecer aqueles que possuem alguma incapacidade funcional que impossibilita a manipulação do teclado e mouse.

A quantidade de idosos que participou do exercício de interação ainda é pequena para justificar conclusões mais precisas. Por este motivo, pretende-se ampliar o número de idosos participantes nas próximas edições da oficina. Apesar disso, a qualidade da interação ocorrida e os depoimentos dos idosos sobre o exercício, são indicadores de que o uso de comandos de voz é uma alternativa viável de interação e que deve ser melhor explorada.

A escolha do software IBM Via-Voice se deu em função de sua disponibilidade no momento de realização da oficina. Percebeu-se a necessidade de realizar um levantamento do estado da arte dos softwares que possibilitam a interação através de comandos de voz, selecionando o mais adequado para o trabalho que se deseja realizar.

Pretende-se desenvolver um ambiente, onde os idosos possam trocar mensagens e participar de discussões através da Internet, formando uma comunidade virtual. Deseja-se que a interação neste ambiente ocorra por meio de comandos de voz. Esta idéia foi divulgada aos idosos que realizaram uma votação para escolha do nome da comunidade. O nome escolhido foi Nossa Idade.

A construção do espaço de discussão está em desenvolvimento e poderá ser consultada através do endereço <http://www.lsc.inf.ufsc.br/nossaidade>.

4. Agradecimentos

A todos os idosos voluntários que participaram das oficinas e que auxiliaram significativamente no desenvolvimento deste trabalho. Aos professores da Universidade Federal de Santa Catarina – UFSC Antônio Carlos Mariani, Edla Maria Fausto Ramos e Walter de Abreu Cybis pela disponibilidade, apoio e opiniões críticas que contribuam para melhoria do trabalho.

5. Referências Bibliográficas

- IBGE 2001. Censo 2000. Disponível em <http://www.datasus.gov.br> (visitado em Junho de 2001)
- Nielsen, J. Designing Web Usability: The Practice of Simplicity. Indianapolis, IN: New Riders Publishing, 2000.
- Mattos P. Cognição e envelhecimento: diagnóstico diferencial pelo exame neuropsicológico. In: Câmara V.D. et al (org). No envelhecimento... o que queremos? Sociedade Brasileira de Geriatria e Gerontologia Seção Rio de Janeiro. Collectanea Symposium, Frôntis Editorial, Rio de Janeiro, 1999.

Análise ergonômica do estado da arte como facilitadora do processo de especificação de IHC : o caso de um ambiente de autoria em RV

Walter de Abreu Cybis¹, Vera R. N. Schuhmacher², Kelly Cristina Goedert³

Univ. Federal de Santa Catarina

¹Departamento de Informática e Estatística

²Curso de Pós Graduação em Ciência da Computação

³Curso de Bacharelado em Ciência da Computação

{cybis, vera} @inf.ufsc.br, kelly@netlan.net

***Abstract.** This paper presents an ergonomic approach for the usability and user interface specification of an interactive system that intends to modify deeply the way to carry out the task that it supports. The ergonomic analysis of the state of the art is proposed as an essential part of this approach, as it allows the identification, with a considerable advance, of problems and qualities of the solution that could be developed in the project.*

***Resumo.** Este artigo trata de uma abordagem ergonômica para a especificação de usabilidade e da interface com o usuário de um sistema interativo que pretende modificar profundamente a maneira de realizar a tarefa que apoia. A análise ergonômica do estado da arte é proposta como parte essencial desta abordagem, na medida em que permite identificar, com uma apreciável antecedência, problemas e qualidades de soluções que poderiam ser desenvolvidas no projeto.*

1. Introdução

Um dos princípios básicos da abordagem ergonômica para o desenvolvimento de sistemas de trabalho, sejam eles informatizados ou não, é o de "conhecer para modificar" uma realidade de trabalho [3]. É preciso conhecer as exigências atuais e futuras pois a introdução de uma nova ferramenta terá um impacto maior ou menor, dependendo da compatibilidade entre as novas e as velhas exigências sobre o usuário e suas formas de realizar o trabalho.

A análise e especificação de sistemas inovadores é dificultada pela aparente inexistência de um quadro de comparação entre o que é e o que será o contexto de realização da tarefa. Frequentemente, este fato é citado como justificativa para a não realização de uma boa análise do existente [6]. Entretanto, e antes de negar a pertinência da análise do trabalho, os autores do método MUSE propõem, que se busque analisar a tarefa com sistemas relacionados ao sistema pretendido (extended systems). De fato, entrevistar, observar e analisar, usuários de sistemas representando o estado-da-arte do projeto é uma estratégia válida para conhecer, com uma apreciável antecedência, as vantagens e inconvenientes de lógicas de operação que poderiam ser adotadas no novo sistema.

2. Definição de escopo do projeto Museu Virtual

As atividades relatadas neste artigo se referem a especificação da usabilidade e da interface com o usuário de um ambiente de autoria de mundos virtuais destinado a estudantes de segundo grau, que através dele, realizarão os trabalhos de pesquisa solicitados pelos professores de diversas disciplinas, como ciências, história, geografia, matemática [11]. Estes trabalhos serão "museus virtuais" construídos pelos estudantes, de forma colaborativa através da Internet, e serão por sua vez, visitados, também através da Internet, por colegas de outras turmas, de outras escolas, em outras cidades ou países. Este ambiente está sendo construído em cooperação entre a Universidade Federal de Santa Catarina, a Universidade Federal do Rio Grande do Sul e a Universidade Federal de São Carlos, que dividem as responsabilidades sobre os aspectos computacionais (UFSC e UFSCar), ergonômicos (UFSC) e pedagógicos (UFRGS) do projeto.

Esta definição de escopo aponta para um sistema inovador, no qual as atuais e futuras formas de executar a tarefa são bastante diferentes. Nestes casos, aplica-se em particular, a análise ergonômica dos contextos atual e futuro de realização da tarefa.

3. Análise ergonômica da tarefa atual e futura.

3.1 Análise da situação atual

Objetivo geral da tarefa de referência é de aprender algo, sobre geografia, história, arte, matemática, ciências, por meio da elaboração e visita a trabalhos de pesquisa escolares. Os alunos participantes do estudo-piloto são habitantes de centros urbanos, pertencem a famílias de classe média. Como adolescentes, os alunos apresentam níveis elevados de ansiedade e de curiosidade. Eles têm uma atitude positiva e bastante facilidade em aprender a utilizar programas de software e esperam um software rápido, em tempo de resposta, e com recursos elaborados, tanto em termos das funções que venha a oferecer como da sua forma de operação (altamente interativo).

3.2 Análise de uma situação futura (com o estado da arte)

Para esta análise realizou-se dois tipos de atividades; uma avaliação heurística das características de diversos sistemas de autoria em 3D existentes no mercado e uma observação de sua utilização por estudantes pertencentes a população-alvo do sistema.

A avaliação heurística foi realizada por dois especialistas em usabilidade que identificaram os problemas e as boas soluções do projeto das interfaces dos ambientes de concepção World Studio, Do 3D e Vr Creator. As principais observações anotadas se referem a problemas relacionados aos seguintes aspectos:

Alta densidade funcional	Manipulação de grupos deficiente
Alta densidade de informação	Passagem de modos demorada e custosa
Má organização das bibliotecas de objetos	Má organização dos comandos de navegação
Denominações muito técnicas	Operação difícil dos comandos de navegação
Manipulação de arestas e pegadas deficiente	

Durante os ensaios de interação, as tarefas de criação e manipulação solicitadas foram realizadas sem problemas pelos estudantes. O modo de operação da interface do programa navegador foi facilmente assimilado pelos usuários, demonstrando ser intuitivo. Nenhum participante teve dificuldades para se deslocar no mundo criado para os ensaios.

4. Requisitos para a IHC do sistema de autoria Museu Virtual

Com base nos resultados das análises ergonômicas realizadas, foi possível definir a seguinte especificação funcional e não funcional geral da usabilidade o sistema:

Quesitos funcionais gerais ligados ao domínio da aplicação do sistema	Quesitos gerais sobre o modo de operação das funções do sistema
Prever o desenvolvimento de uma ferramenta simples, com poucas funções	Fornecer estilos de interface compatíveis com os ambientes/programas/dispositivos mais empregados atualmente pelos estudantes.
Prever chamadas para editores para textos, de gráficos, de sons e de vídeo.	Fornecer boa condução ao usuário novato
Prever chamadas para ferramentas de apoio a discussões síncronas e assíncronas.	Organizar as bibliotecas de modo a evitar a confusão entre categorias de recursos distintas.
Prever recursos para o trabalho compartilhamento de recursos.	Fornecer flexibilidade em termos de diversas maneiras ou estilos de interação para realizar uma mesma tarefa.
Quesitos funcionais gerais ligados a interface do sistema	Minimizar a carga de trabalho para o uso da ferramenta.
Prever facilidades de personalização.	Definir o vocabulário da interface por meio do envolvimento com o usuário.
Fornecer uma adequada gestão de erros	Definir pegas, arestas e cursores fáceis de usar
Prever um site de ajuda, além das facilidades de comunicação com os colegas.	Autorizar a manipulação direta de grupos de objetos.
Prever compatibilidade em termos dos dispositivos de interação: apenas o mouse e o teclado.	Definir um modo de navegação baseado na atuação somente sobre o botão de controle associado ao movimento desejado.
Prever cursos de treinamento do ambiente Museu Virtual.	

5. Discussões

A questão a ser analisada se refere a influência do estado da arte sobre as soluções do projeto. De fato, as soluções do projeto se assemelham ao estado da arte por que os ambientes avaliados provaram ter boa usabilidade. Por que não reutilizar algo que funciona comprovadamente bem? Nos casos em que as soluções propostas pelo estado da arte fossem de má qualidade, a estratégia ainda seria válida pois ela impediria a reutilização de mas soluções e uma nova interface deveria ser modelada e testada junto aos usuários, o mais cedo o possível.

6. Bibliografia

- [3] Cybis W.A., P. M. S., Silveira M.C, Gamez L. (1998). Uma abordagem ergonômica para o desenvolvimento de sistemas interativos. IHC 1998, Maringá.
- [6] Lim, K. Y. (1996). "Structured task analysis: an instantiation of the MUSE method for usability engineering." *Interacting with Computers* 8(1): 31-50.
- [11] Wazlawick, R. S., Rosatelli, M. C., Ramos, E. M. F., Cybis, W. A., Storb, B. H., Schuhmacher, V. R. N., Mariani, A. C., Kirner. T. G., Kirner C. & Fagundes, L. C. (2001). Providing more interactivity to virtual museums: A proposal for a VR authoring tool. *Presence: Teleoperators and Virtual Environments*, 10 (6), a ser publicado.

Ctrl-P: Enhancing the Readability and Comprehension of Printed Academic Materials

Mario Alberto Moreno Rocha

Instituto de Electrónica y Computación, Universidad Tecnológica de la Mixteca, Km 2.5
Carretera a Acatlima, Huajuapán de León, Oaxaca, CP 69000, México

mmoreno@mixteco.utm.mx

***Abstract.** Distance learning offers unparalleled advantages when compared to the traditional education. However, studies shows that a big majority of students download their materials and print them out for study off-line, thus reducing the extraordinary multimedia experience into the old and traditional printed paper sheet. How could the design of academic contents be achieved effectively having in mind the “Ctrl+P” factor? The poster will discuss the possible way in which printed content would benefit from HCI studies, a comparison between paper and reading devices and recommendations to virtual educators for the creation of better printed materials.*

1. Introduction

Since the dawn of the so-called distributed or virtual education, more properly refereed as distance learning, the possibilities looked endless. Countries who cannot compete in the new world order due by their limited resources find in the distance learning a vehicle to leap ahead on their educational drawback.

At the Universidad Tecnológica de la Mixteca, such important endeavour is present by the Virtual University programme (UV) [Universidad Virtual, 2001]. Founded in 1998, the UV offers now two different postgraduate study programmes. The number of enrolled students duplicates every time. Our students are scattered all across the country, but in the future, students from all over Latin America are expected to participate.

Despite the efforts to enrich the academic contents on-line and make education more attractive, a recent survey shown that a big majority of our students print out the downloaded materials for study off-line, then reducing the all multimedia experience into a traditional, old fashioned printed paper sheet.

This tendency cannot be reverted. How could HCI be usefully applied to improve the design of printed materials in order to enhance readability and comprehension of academic content?

2. Distance Learning Analysis

Traditional education, based on the classroom-and-teacher scheme, has proved effective for more than 200 years. However, it could be enhanced through new technologies. With the easy access to personal computers and its increased multimedia capabilities, its utilisation in education was a natural consequence [Moreno-Rocha, 1998].

The ubiquitous communication technology of the Internet, the intuitive presentation of information, and the wonders of the World Wide Web offers a possibility to find a model

where it could take the best of both methods and obtain a new model; the Web based distance learning education.

However, it is a widespread myth that "The content, value, and quality of information and knowledge are improved simply because is offered in multimedia or over the Web" [Fischer, 1997].

3. The Ctrl+P Factor

Printing technology has 500 years history of improving, compared to the 40 years or so of screen developments. Paper surface and colour, typeface, character width, letter sharpness, text contrast with paper, margin sizes, spacing between the lines, and even room lighting have been explored in efforts to produce the most appealing readable format [Shneiderman, 1998]. On the other side, computer screens cause visual fatigue and stress. The comprehensibility studying from displays may be below to that obtained by reading from printed materials.

Our reading speed diminishes a 25% when reading from a screen compared to that from a printed source [Nielsen, 1999]. It cannot be denied the extraordinary development is displays. It is just a matter of time when screens and paper will share the same characteristics, but for the time being, paper is preferred over any computer display.

With printed materials there is no usability problems. The ability to turn the page or follow a jump line into a inner page is practical universal: you do not need to teach the reader these skills [Outing, 2000]. There is no resolution problem as well, it does not faint on the eyes.

3.1 Paper v. Screen

Information being available, however, does not necessarily mean being readily accessible, and users, or potential users, of information which is available often experience frustration in locating, accessing, and interpreting the information they want [Nickerson, 1995].

We are in a transitional phase. Although we use a word processor to write texts, we still print it and read it from paper [Driel, 1999]. This is the case of academic contents. Lessons are published on the Web but they are not suitable to be studied on the screen.

The disadvantages of reading from displays are summarised in the list below: [Shneiderman, 1998].

Poor fonts, specially in low resolution monitors.

Low contrast, and fuzzy characters boundaries.

Obtrusive emitted light, screen flicker and a possible curved display surface.

Smalls displays, requires disruptive page turning controls.

Layout and formatting, inappropriate line width or awkward justification.

Reduced hand and body motion, compared to paper, may cause fatigue.

Unfamiliarity of displays, and the anxiety that the image may disappear.

According to Nielsen [Nielsen, 1999] readers on electronic devices "scans" the information, instead of careful reading, thus the information must be formatted in such a

way [Henning, 1998]. The poster will display an example and will explain how this could be achieved.

4. Enhancing Readability and Comprehensibility

The recommended solution is to create two versions of the same content [Nielsen, 1999]. One, formatted into Web pages and prepared to be "scanned" instead of read, complete with hyperlinks and multimedia. The other should be an optimised version for printing. This one should be one long file, preferably on PostScript or PDF format, which could be downloaded, printed out and read comfortably.

The poster will feature a comparison between paper readability and various reading devices, namely a PC screen, laptop display, PalmPilot and an eBook.

5. Conclusion

"Computers are great to store information, but they are not that good to register sudden ideas, thoughts, feelings. For that, paper is still quite superior, you can take it, fold it, put it in your pocket and revise again, when convenient" [Fayyad, 1999].

Academic designers work enormously on the design of academic web pages, but the printed paper issue surpasses those efforts every time. More attention should be taken into consideration when developing academic content to be printed.

References

- Driel, Hans van: Digitaal communiceren, <http://cwis.kub.nl/~fdl/general/people/drielhv/lezing/kubweb/>
- Fayyad, Usama, Microsoft Research, cited in "The Power of Writing", National Geographic en Español, August 1999, 110-132
- Fischer, Gerhard: Life Long Learning, More Than Screen Deep: Toward Every-Citizen Interfaces to the Nation's Information Infrastructure, 1st Ed, National Academy Press, Washington DC, (1997), 382-388
- Henning, Kathy: Writing for Readers Who Scan, <http://clickz.com/article/cz.3326.html>
- Moreno-Rocha, Mario, A: Metodologías de enseñanza, Universidad Virtual, UTM, 1998
- Nickerson, Raymond, S (editor): Emerging Needs and Opportunities for Human Factors Research, 1st Ed, National Academy Press, Washington DC, (1995) 38-42
- Nielsen, Jakob: In Defense of Print, <http://www.useit.com/alertbox/9602.html>
- Outing, Steve: Reader Usability Wisdom From a Web Guru, <http://www.mediainfo.com/ephone/news/newshrm/stop/st070297.htm>
- Shneiderman, Ben: Designing the User Interface: Strategies for Effective Human Computer Interaction, 3rd Ed, Addison Wesley, USA (1998) 410-441
- Universidad Virtual, UTM, <http://virtual.utm.mx/>

Desenvolvendo Sistemas Interativos com a UML segundo o Princípio de Independência do Diálogo Humano-Computador

Elizabeth Furtado, Régis Simão

Universidade de Fortaleza – Mestrado em Informática Aplicada – Washington Soares,
1321 – Bairro Edson Queiroz – Fortaleza (Ceará), 85-4773268
elizabeth@feq.unifor.br, regis.simao@serpro.gov.br

Abstract. *This work is focused on describing a new method to develop interactive systems (IS). This method is centered to user and follows the principle of human-computer dialogue independence, in order to improve IS usability. This dialogue, which represents the user activities, is modeled through packages and defined taking into account ergonomic principles. The presented method aims to guide software engineers in HCI (Human-Computer Interaction) design using UML (Unified Model Language).*

Resumo. *Este trabalho apresenta um novo método de desenvolvimento de sistema interativo (SI). Este método está centrado no usuário e respeita o princípio de independência do diálogo humano-computador, a fim de melhorar a usabilidade do sistema desenvolvido. Este diálogo, o qual representa as atividades do usuário, possui uma modelagem orientada a objetos através da integração de pacotes levando em consideração tal princípio. Este método objetiva guiar os engenheiros de software a projetar interfaces através da UML (Unified Model Language).*

1. Introdução

Por sistema interativo (SI), entende-se um conjunto de programas manipulado pelo usuário cuja realização de suas funções visa resolver um determinado problema. Um SI é composto de uma parte não interativa (aplicação) e de uma parte interativa (interface). A aplicação é constituída de dados, que representam a parte estática do sistema (como, as entidades) e de tratamento sobre estes dados (lógica da aplicação). A interface contém objetos interativos e módulos responsáveis pelo controle do diálogo homem-computador (tais como, de tratamento de eventos, de manutenção da coerência entre os objetos interativos e os dados da aplicação).

Problemas de desenvolvimento e de manutenção de um SI são, em grande parte, devido à má estruturação da modelagem das especificações (dados e funções de um sistema). Para atenuar estes problemas, vários métodos de desenvolvimento apoiam esta modelagem, tais como: o método unificado definido pelos autores da UML e a modelagem CRC. No entanto, esses métodos falham por não especificarem claramente como respeitar o princípio da independência do diálogo [Lafon1991], isto é, como modelar a aplicação de forma independente da interface e do controle do diálogo. Outro problema estudado em engenharia de software e áreas afins diz respeito à usabilidade de um SI. De maneira geral, os estudos em IHC (Interação Homem-Computador) visam integrar fatores humanos (recomendações ergonômicas, conceitos relativos ao comportamento de um indivíduo) no

processo de desenvolvimento de um SI, afim de que as interfaces se adaptem melhor às necessidades do usuário [Furtado1997].

2. O Método Proposto

O método, apresentado na figura 1 e elaborado para o desenvolvimento de SI, combina vários ciclos de vida, pois, baseia-se na construção de protótipos e no desenvolvimento incremental e iterativo. Este método define a UML como linguagem para expressar as decisões e soluções. A figura 1 ilustra as fases gerais do método colocando em destaque, na parte superior, o processo de desenvolvimento correspondente a Aplicação e, na parte inferior, o processo correspondente a Interface. A ilustração não pretende definir a seqüência a ser seguida entre as fases, a qual depende do ciclo de vida em que se baseia o método.

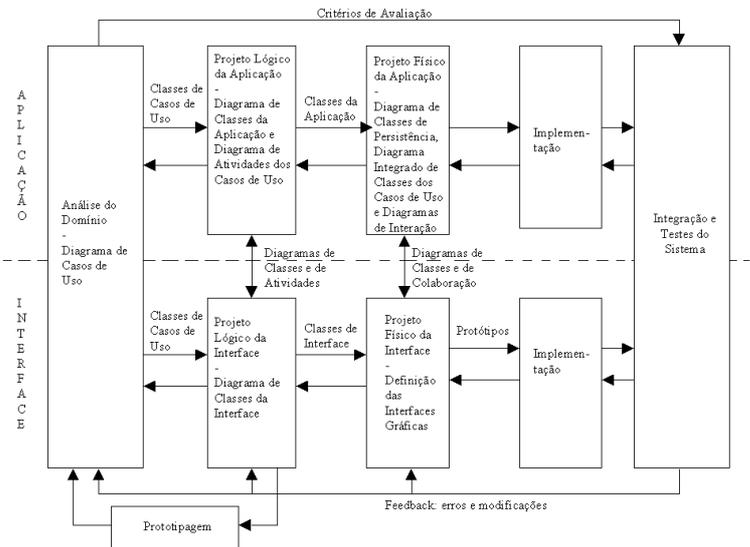


Figura 1: O método elaborado.

Na fase de Análise do Domínio, deve-se identificar e modelar as necessidades do usuário e o seu relacionamento com o sistema existente. Cenário é uma técnica usada para expressar tais objetivos, porque representa situações reais, que os usuários costumam se defrontar para atender suas necessidades. A generalização destas situações está associada à noção de caso de uso descrita em UML [Quatrani1998]. O resultado desta fase é a validação do diagrama de caso de uso construído com o usuário. Neste momento, o engenheiro de software está apto a fazer também um glossário da aplicação e uma estimativa de custos e de cronograma.

O Projeto Lógico da Aplicação visa descrever o sistema a ser desenvolvido, descrevendo as primeiras classes que estão presentes no domínio existente, e detalhando os casos de uso modelados, definindo suas atividades principais. Nesta fase, somente as classes que pertencem ao domínio existente, chamadas *classes da aplicação*, são identificadas. Sugere-

se que elas sejam agrupadas num pacote específico da aplicação. Um caso de uso é descrito através de um conjunto de atividades, que se referem às ações elementares que devem ser executadas pelo sistema ou pelo usuário. Embora existam, na literatura, vários modelos de representar as atividades, é adotado o diagrama de atividades da UML. Outros diagramas da UML (como diagrama de estado e de seqüência) podem também ser modelados neste momento.

No Projeto Lógico da Interface, o projetista de interface deve verificar para cada caso de uso que informações estarão disponíveis para o usuário durante a realização de uma determinada atividade. Em seguida, ele deve especificar a interface definindo seus componentes de interação. Tais componentes são: espaços de interação (EIs), que são os lugares de atividade virtual que oferecem ao usuário a possibilidade de navegar para outro espaço e/ou manipular os dados contidos nos espaços; e objetos de interação (OIs), que são os dados manipulados. Heurísticas e regras ergonômicas, descritas em [Furtado1999], relativas às questões de usabilidade, foram refinadas para ajudar o projetista na definição de EIs e seus OIs a partir de casos de uso e suas atividades. Os EIs e OIs são representados através de classes de interface. O resultado desta fase é a geração do diagrama de classe no pacote da interface e de protótipos. Durante a validação dos protótipos, qualquer retificação implica em voltar às fases anteriores, garantindo o caráter iterativo do método.

Na fase Projeto Físico da Aplicação, o resultado da análise e projeto lógico da aplicação é expandido em soluções técnicas através de: definição do pacote de persistência, utilização de bibliotecas da linguagem de programação escolhida, detalhamento dos métodos das classes e refinamento do diagrama de classe do pacote de interface. Quanto a esta última tarefa, é importante lembrar que as fases do processo em estudo não são realizadas de forma seqüencial. Assim, são apresentadas, nos diagramas de classes, informações provenientes de decisões tomadas na fase de projeto físico da interface. Esta fase consiste na geração de um diagrama de classe integrado para cada caso de uso. Esta integração é feita entre as classes dos pacotes de aplicação, de persistência e de interface. O objetivo é que tais diagramas permitam uma visão melhor dos fatores que podem influenciar uma decisão associada a um caso de uso. Nesta fase, modela-se o comportamento dinâmico dos casos de uso, utilizando-se os diagramas de interação da UML: diagrama de seqüência e/ou colaboração.

No Projeto Físico da Interface, o projetista faz um estudo do sistema sob o ponto de vista interativo, para definir os aspectos visuais das interfaces e definir como o sistema tratará os eventos vindos da interface ou da aplicação. Para realizar estas operações, sugerimos que o projetista aplique algumas recomendações ergonômicas para ajudá-lo, por exemplo, selecionar os objetos interativos, definir seus atributos gráficos (por exemplo, cor, tamanho e posição de um objeto interativo na tela), o estilo de interação (se baseado em menu ou em comando), etc. Como resultado desta fase, obtém-se as telas gráficas do sistema.

As fases de Implementação visam a construção ou implementação real das classes. A fase de Teste envolve a avaliação do SI completo, realizando testes integrados da aplicação e da interface.

3. Conclusões

Neste trabalho, apresenta-se um método de desenvolvimento de SI em que propõe-se a modelagem de aspectos da aplicação de forma independente da modelagem de aspectos

interativos. De forma separada, estas modelagens são feitas em diagramas de classes por pacotes e se integram na construção dos diagramas de classe por caso de uso e na construção de diagramas de atividades e de interação. Para elaborar este método, aplicou-se uma metodologia de trabalho interdisciplinar especificada pelo envolvimento integrador das disciplinas trabalhadas, ergonomia cognitiva, engenharia de software e IHC.

Pôde-se observar claramente a separação dos elementos de aplicação e de interface, o que facilitou muito o tratamento dos detalhes específicos de cada aspecto independentemente. Observou-se também que manutenções foram mais facilmente localizadas e com maior visibilidade dos impactos das mudanças.

Referências

- [Furtado1997] Furtado, E.S. "Mise en oeuvre d'une méthode de conception d'interfaces adaptatives pour des systèmes de supervision à partir des Spécifications Conceptuelles". Thèse de doctorat. França. Março. 1997.
- [Furtado1999] Furtado, E.S. "An Approach to improve design and usability of user interfaces for Supervision systems by using Human factors", HCI' 99 International, Munich, Germany. August 1999.
- [Hix1993] Hix, D., Hartson, R. "Developing user interfaces". Wiley. 1993.
- [Lafon1991] Lafon, M. "Interfaces Homme-machine : Vue d'ensemble et perspectives". Actes du congrès Génie logiciel & Systèmes Experts, Interfaces homme-machine, Maquettage & prototypage. Vol. 24, pp. 4-16. 1991.
- [Quatrani1998] Quatrani, T., Booch, G. Visual Modeling With Rational Rose and Uml. Addison-Wesley Object Technology Series. 1998.

Ergonomia da Informação: apontamentos para *webdesigners*

Anna Lúcia dos Santos Vieira e Silva

Rua São João, 491
13416-810 - Piracicaba, SP - Brasil
lilu@terra.com.br

Resumo: Alguns pontos de vista sobre ergonomia da informação são expostos para suscitar questões específicas ao *webdesigner*. Objetivos, organização, clareza e coerência são enfocados como fatores que facilitam a compreensão e transmissão de conhecimentos e a experimentação, como necessidade para o desenvolvimento da linguagem hipermídia.

Questões contemporâneas implicam um paradoxo. Embora seja necessário um distanciamento analítico e crítico, estamos tão imersos como coniventes a tudo que é atual, direta ou indiretamente. Os dias de hoje não estão congelados em um instante presente. A mutabilidade traz consigo cicatrizes e sortilégios históricos, desejos e tendências.

Fazemos parte de um sistema geral de trocas de matéria e conhecimento que nos lança num fluxo de atualidade incansavelmente questionável. As informações são tantas e de tantas origens que já não discernimos o que é informativo, para que fim, ou qual é a pertinência dos conteúdos que se apresentam e que são buscados. Vivemos um bombardeamento de informações sem precedentes. Como atuar neste contexto?

Quanto mais instigados nos fazemos mais descobrimos a amplitude da nossa ignorância e o alcance do nosso conhecimento. Externamos e estendemos nossos sentidos quando ampliamos nossa capacidade de comunicação. A Ergonomia de Informação, surge para auxiliar o *designer* na organização das informações que manipula, tornando-as claras e de fácil acesso. Mais uma vez a pergunta: como?

O primeiro apontamento para *webdesigners* diz respeito aos objetivos pessoais, profissionais e de cada projeto específico. A pergunta passa de **como?** para **o que?**. O objetivo pessoal deve estar de acordo com as capacidades, intenções, interesses, oportunidades e necessidades particulares. Quanto mais o *designer* adquire autonomia sobre o repertório que lhe concerne, mais próximo fica de acertar a coerência entre o que deseja e o que produz. O objetivo profissional deve corresponder a sua postura ética e estética. Para cada projeto específico, o contexto, as intenções do cliente e o público alvo devem ser relevados. O risco de não haver objetivos é o de correr em círculos, sem sair do lugar.

O objetivo desse texto, por exemplo, é reduzir, ou melhor, traduzir o que consideramos mais importante a ser questionado e apontado sobre um assunto tão recente quanto abrangente.

Em um sentido geral, todos os fatores que **facilitem** a compreensão e transmissão de conhecimentos são de competência da **ergonomia de informação**. Ela estuda a organização e sistematização das informações, conforto visual, composição dos elementos gráficos, de modo que a mensagem seja transmitida com eficiência. Questões como nível

de contraste, tamanho de letra e de texto, uso de cores, fotografias, legendas, gráficos e animações são analisadas funcionalmente. Funcionar nesse caso equivale a objetividade, clareza e facilidade de encontrar o que se procura.

Muitas crianças que cresceram com *videogames* e se familiarizaram com a linguagem e velocidade infográfica sabem manipular as ferramentas do sistema. Isto não implica necessariamente em uma concretização de objetivos, tampouco caracteriza um *designer*. O *webdesigner*, por compreender com mais profundidade a linguagem específica do computador e ser invariavelmente um usuário, pode se valer da ergonomia de informação para alcançar um objetivo. Agora sim: como?

Não vamos propor um método, e sim apresentar duas áreas afins. A **arquitetura de informação** estuda a estrutura, conexões, hierarquias e afinidades das informações para dispô-las na rede que configura a Internet. Para isso, o *webdesigner* necessita de um levantamento das informações necessárias para organizá-las. Uma boa **pesquisa** sobre o assunto concernente aos nossos interesses aumenta as possibilidades de eleição e combinação das informações. O **desenho de interface** fornece o canal de ligação entre a linguagem do computador e a humana, fazendo com que bits tornem-se palavras, imagens, movimento, sons, comandos e links.

Quando o que construímos designa estética, funcionalidade e comunicação, normas e regras são provisórias ou insuficientes, quando não ambas. O contexto é mutável e complexo.

A ergonomia procura facilitar a clareza, organização, acessibilidade, estrutura, navegabilidade e fazer com que o usuário encontre as informações que procura da maneira mais rápida e eficaz. Por outro lado, artistas e *designers* experimentam novas possibilidades de uso do computador, buscam o inesperado, aguçam a sensibilidade e as percepções, propõem interpretações variadas.

Arte e ergonomia, mesmo com divergências de objetivo ou de método, coexistem e são necessárias para o desenvolvimento da nova linguagem. São necessárias uma para outra. A investigação criativa amplia as possibilidades de uso enquanto a ergonomia organiza e estrutura informações que podem ser apropriadas no processo artístico.

Devemos olhar para as coisas como processos e não como modelos.

“Analisando assim as coisas, o modelo dos modelos almejado por Palomar deverá servir para obter modelos transparentes, diáfanos, sutis como teias de aranha; talvez até mesmo para dissolver os modelos, ou até mesmo para dissolver-se a si próprio.”

Italo Calvino (1990: 99)

Bibliografia citada:

CALVINO, Italo. (1990) *Palomar*. São Paulo: Companhia das Letras.

JOHNSON, Steven. (1997) *Cultura da Interface: como o computador transforma nossa maneira de criar e comunicar*. Trad. Maria Luiza X. de A. Borges. Rio de Janeiro: Jorge Zahar Editor.

GOMSQ: Um método de modelagem de tarefas para sistemas educacionais baseado em comunicabilidade

**Catharine Ferreira Bach, Raquel Oliveira Prates,
Rosa Maria Videira de Figueiredo**

Instituto de Matemática e Estatística – Universidade do Estado do Rio de Janeiro
catharinebach@yahoo.com, {raquel, rosa}@ime.uerj.br

***Abstract.** In this work we present GOMSQ model method of task for an educational system of help to the SIMPLEX method teaching that serves as base in the GOMS simplified and in communicability. The purpose of this method permits the teacher to foresee the doubts that the students have concerned with the matter and with this the interface designer can design a solving that have the answers to it trying to foresee the breakages in the process of learning of the students.*

***Resumo.** Neste trabalho apresentamos um método de modelagem de tarefas, GOMSQ, para um sistema educacional de apoio ao ensino do método SIMPLEX que se baseia no GOMS simplificado e em comunicabilidade. O objetivo deste método é permitir ao professor prever as dúvidas relacionadas ao conteúdo que os alunos porventura possam vir a ter, e com isso permitir que o projetista de interface projete uma solução que inclua as respostas para elas, tentando assim prevenir rupturas no processo de aprendizado dos alunos.*

1. Introdução

O projeto de uma interface é extremamente importante na construção de um sistema, porque é através da interface que o usuário entende e interage com o sistema. Projetar uma interface educacional é um desafio ainda maior. Uma interface educacional, assim como qualquer interface genérica, deve ser projetada para preencher os requisitos de usabilidade e comunicabilidade inerentes a qualquer projeto de interface. No entanto, uma interface educacional deve também proporcionar o aprendizado de um conteúdo pelo usuário (no caso, um aluno) [2, 5].

De acordo com Martins et al.[3], na abordagem da Engenharia Semiótica, a interface educacional é vista como uma mensagem conjunta entre o educador e o projetista de interfaces para o usuário. O educador torna-se responsável por definir o conteúdo a ser ensinado e como o software deve apoiar o ensino desse conteúdo. Cabe ao projetista estabelecer uma solução para transmitir a mensagem desejada pelo educador ao aluno, ou seja, definir como o aluno deve interagir com o software para aprender o conteúdo desejado.

Na UERJ (Universidade do Estado do Rio de Janeiro), o professor de Programação Linear reparou ao ensinar o Método Simplex que a grande quantidade de operações algébricas necessárias para resolver um problema de grande porte pode desmotivar o aluno a aprender o método ou fazer com que ele priorize as operações algébricas em detrimento ao entendimento do processo. Assim, este professor identificou a necessidade de ter um

instrumento, de caráter educacional, capaz de motivar seus alunos no processo de aprendizado do Método Simplex. A solução prevista pelo professor foi a de fornecer aos alunos um software que lhe permita estudar o Simplex e ampliar sua capacidade de abstração, sem a preocupação excessiva com as operações algébricas. Além disso, seria desejável que além de dar apoio ao aprendizado da execução do Método Simplex, o software também apoiasse o entendimento do mesmo.

Apresentaremos, a seguir, um método de modelagem de tarefas baseado no GOMS simplificado e no método de avaliação da comunicabilidade de interfaces [4] para o desenvolvimento de uma interface educacional, que abrange os dois níveis de apoio ao aprendizado desejado pelo professor.

2. Prevenindo rupturas do aprendizado

O método de avaliação de comunicabilidade de uma interface associa às ações dos usuários um conjunto de interjeições, que o usuário poderia exprimir durante a execução de uma tarefa, e que identificariam rupturas da sua interação com a aplicação.

Para interfaces educacionais, a mensagem a ser transmitida deve incluir o conteúdo a ser aprendido. Assim, nosso objetivo foi criar um método que guiasse o design da interface do sistema AMPLIAR (**A**Mbiente para Aprendizado em **P**rogramação **L**ine**A**R) de forma a tentar prevenir as possíveis rupturas no aprendizado do conteúdo. O método consiste em (1) fazer a modelagem de tarefas do sistema utilizando o GOMS simplificado (Lee, 1993), (2) a cada meta identificada na resolução do problema, acrescentar as interjeições que identificariam as dúvidas que o aluno poderia ter naquele passo da resolução e (3) associar a cada interjeição, a cada passo, as respostas fornecidas pelo professor. Nos referiremos ao método utilizado como GOMSQ (*G*oals, *O*perators, *M*ethods, *S*election Rules and *Q*uestions).

Assim, a partir das possíveis dúvidas às quais o sistema deverá dar apoio durante a execução do Método Simplex definimos o conjunto de interjeições que identificam o tipo de pergunta que o aluno poderia fazer ao sistema a cada etapa do método. O conjunto de interjeições obtido é apresentado a seguir:

- **O que é?:** O aluno quer saber o que é determinado elemento que aparece na resolução do Simplex;
- **E agora?**¹: O aluno quer saber qual o próximo passo para a resolução do Método Simplex;
- **Como?:** O aluno quer saber como ele deve fazer para executar uma determinada tarefa durante a resolução do Simplex;
- **Por quê?:** O aluno quer saber por que determinado elemento aparece, por que determinado elemento possui o valor apresentado ou por que ele deve executar o passo indicado pelo software para prosseguir na execução do Simplex;
- **Eu entendi?:** O aluno quer saber se realmente entendeu o que está fazendo, justificando a sua decisão;

¹ Embora as interjeições O que é? e E agora? também pertençam ao conjunto de interjeições original para a avaliação da comunicabilidade de uma interface, aqui elas se referem ao conteúdo educacional.

Ao responder estas perguntas, o sistema apóia o aprendizado do aluno em dois níveis. O primeiro nível refere-se ao apoio para a execução do método e associa-se às interjeições O que é?, E agora?, Como?, pois através dessas interjeições o sistema fornecerá apoio ao usuário para a realização dos passos necessários para a execução de uma tarefa. O segundo nível apóia o entendimento do método associado às interjeições Por quê? e Eu entendi?. Nesse segundo nível, Por quê? é identificado quando o aluno precisa de uma justificativa para uma determinada ocorrência durante a resolução; já a interjeição Eu entendi? é expressa quando o próprio aluno deseja justificar o que está fazendo, e precisa apenas de uma confirmação para sua justificativa.

3. A Modelagem de Tarefas Utilizando o GOMSQ

Na Modelagem de Tarefas do sistema AMPLIAR, foi identificada uma meta geral: resolver um problema de programação linear utilizando o Método Simplex. Tal meta, entretanto, pode ser dividida nas seguintes sub-metas: (1) Colocar o problema no formato padrão; (2) Definir uma solução básica viável inicial; (3) Definir qual variável entrará na base; (4) Definir qual variável sairá da base; (5) Apresentar a solução do problema. A seguir, ilustramos como exemplo, a modelagem da tarefa 3 utilizando o GOMSQ:

3: Definir a variável que entrará na base

O que é a variável que entrará na base?

R: É uma variável não pertencente a base atual que ao assumir um valor maior que zero permitirá reduzir o valor da função objetivo.

O que é a base?

R: A base B é uma submatriz da matriz A , com $B \in R^{n \times n}$ e $\det(B) \neq 0$, tal que $x_B = B^{-1}b \geq 0$. A solução associada a tal base é denominada solução básica viável e é um ponto extremo da região viável.

Como faço para escolher a variável que entrará na base?

R: Procure dentre todas as variáveis fora da base uma variável x_k com $z_k - c_k > 0$.

Por que tenho que escolher a variável que entrará na base?

R: Porque a solução básica associada a base atual não é ótima, e é possível diminuir o valor da função objetivo ao fazer uma variável $x_k > 0$, ou seja, ao fazer x_k entrar na base, se $z_k - c_k > 0$.

E agora, o que faço depois de escolher a variável que entrará na base?

R: Escolha a variável que vai sair da base, no lugar da qual esta vai entrar.

Como a modelagem acima é meramente ilustrativa, ela não está completa. Para completar nosso modelo, cada submeta da modelagem da tarefa deve apresentar as interjeições e suas respostas naquele contexto específico, conforme fizemos para a meta 3. Deste modo, para cada uma das demais submetas, assim como para as submetas da meta 3 teríamos que apresentar as interjeições e as respostas previstas pelo professor, em cada caso.

4. Conclusão

Neste trabalho apresentamos a proposta de um método de modelagem de tarefas para sistemas educacionais baseado no GOMS simplificado e em comunicabilidade. Este

método enfatiza a mensagem sendo enviada pelo professor ao aluno através da interface do software, ou seja, os aspectos relacionados ao aprendizado do aluno. Assim, o nosso objetivo era desenvolver um sistema que prevenisse, ou pelo menos minimizasse, as rupturas no aprendizado do aluno.

O sistema AMPLIAR está em fase final de desenvolvimento e será testado no próximo semestre junto aos alunos de programação linear da UERJ. Os nossos objetivos com esta avaliação são (1) verificar se o sistema conseguiu atingir seus objetivos educacionais, ou seja, apoiar o aprendizado e entendimento do aluno do método SIMPLEX ; (2) identificar que uso eles fazem das interjeições disponíveis e de que forma o conjunto de interjeições contribui no processo de aprendizado.

O método proposto, GOMSQ, foi desenvolvido e utilizado para o desenvolvimento do sistema AMPLIAR e deve também ser extensivamente avaliado. Os próximos passos nesta direção incluem investigar se o conjunto de interjeições proposto é suficiente para o tipo de apoio desejado pelo educador e se outros conjuntos de interjeições podem ser definidos para fornecer outros tipos de apoio ao aluno. Além disso, uma questão interessante a ser investigada é se este método facilita a comunicação entre o educador e o projetista de interface durante o design.

Referências Bibliográficas:

- [1] Lee, G. (1993) "Object-Oriented GUI Application Development". NJ: Prentice Hall.
- [2] Lewis, C. & Brand, C. & Cherry, G. & Rader, C. (1998). "Adapting User Interface Design Methods to the Design of Educational Activities". CHI 98, Los Angeles – USA, April, 18-23.
- [3] Martins, I. & Santos, N. & Prates, R. (2000). "Interfaces de Ambientes Educacionais: Um Trabalho Conjunto do Educador e Projetista de Interfaces". Cadernos do IME, série Informática, Vol. 9, Dezembro, 2000, pp 12-19.
- [4] Prates, R.O., de Souza, C.S., Barbosa, S.D.J. (2000) "A Method for Evaluating the Communicability of User Interfaces". *Interactions*. Jan-Feb 2000.
- [5] Soloway, E. & Bielaczyc, K. (1996). "Interactive Learning Environments: Where They've Come From & Where They're Going. "

Logo3D – uma ferramenta de auxílio na aprendizagem de algoritmos

Silvano Maneck Malfatti¹, Deise Juliana Francisco¹,
Maria Augusta Silveira Netto Nunes¹, Selan Rodrigues dos Santos²

¹ URI – Universidade Regional Integrada do Alto Uruguai e das Missões,
Santo Ângelo, RS, Brasil.

²The School of Computing of Leeds

{smalfatti, dfrancis, guta}@urisan.tche.br, selan@comp.leeds.ac.uk

Resumo. *O objetivo deste trabalho é o desenvolvimento do software educacional Logo3D, que é uma extensão tridimensional do LOGO apresentado por Seymour Papert¹. O software é destinado principalmente aos alunos iniciantes no curso de Ciência da Computação, como ferramenta auxiliar no aprendizado de algoritmos.*

1. Introdução

Tendo em vista que uma das principais dificuldades dos alunos iniciantes no curso de Ciência da Computação é a disciplina de Algoritmos e Programação, há a necessidade de determinar mecanismos e metodologias eficientes para o ensino da concepção, projeto e implementação de algoritmos corretos e eficientes.

Estas dificuldades podem ser atribuídas a uma série de fatores como a dificuldade de raciocínio lógico matemático e lógico analítico. Sendo assim, um dos principais obstáculos a ser superado consiste na tradução da solução mental para a algorítmica, expressa através de comandos finitos e bem definidos em uma folha de papel.

O presente trabalho tenta minimizar estas dificuldades atacando diretamente o abismo existente entre a transposição da solução mental para a algorítmica, através da aproximação do aspecto estático e dinâmico do mesmo.

Para atingir tal objetivo lança-se mão da já consagrada linguagem de programação LOGO[1] em associação com as técnicas de computação gráfica tridimensional. A inclusão da característica tridimensional tem como objetivo trabalhar mais uma faceta da intelectualidade do estudante, forçando-o a desenvolver e trabalhar suas habilidades espaciais.

¹ Seymour Papert é o criador do LOGO (Bidimensional), onde seu principal objetivo é fazer com que através de comandos como para frente, para trás, giro a direita e giro a esquerda, possamos deslocar um objeto representado por uma tartaruga através do plano (x , y). Onde os movimentos da tartaruga resultam em figuras geométricas.

2. O Software LOGO3D

A incorporação da característica 3D ao LOGO propicia uma movimentação através do espaço, no qual a tartaruga move-se através dos eixos (x, y , z), e isto faz com que o usuário possa descrever objetos cada vez mais perto da realidade[2].

O usuário interage com o ambiente através de duas formas: a primeira é através do modo interativo em que a movimentação da tartaruga ocorre através do clique do mouse em uma paleta que oferece os comandos básicos de movimentação com valores de transformação pré-estabelecidos; a segunda forma é através do modo texto na qual é possível escrever o comando a ser utilizado assim como o valor da transformação.

No modo texto, além dos comandos básicos de movimentação o usuário conta também com estruturas de blocos, estruturas condicionais, estruturas de repetição e subrotinas que dão suporte a programação.

No modo interativo a solução de determinado problema pode ser realizada passo a passo. No modo texto utiliza-se estruturas de controle para “ ensinar ”[3] o computador a resolver o problema.

Vale a pena salientar que como o LOGO3D conserva as características básicas do LOGO, ele também permite ao usuário definir procedimentos e conseqüentemente oferece suporte a recursão.

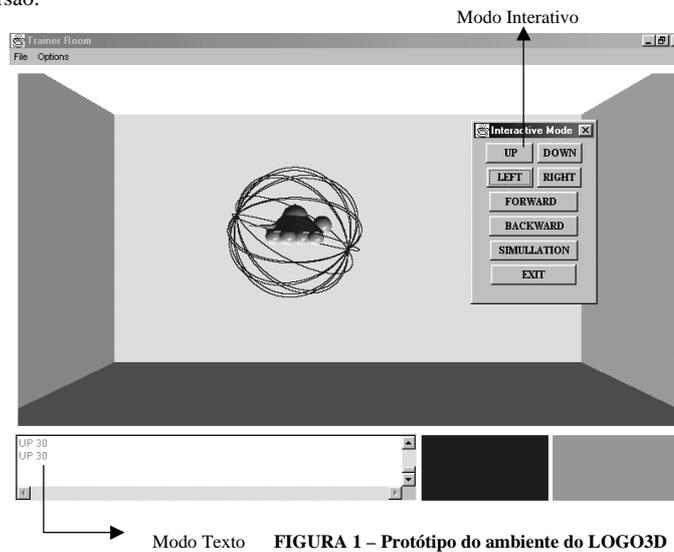


FIGURA 1 – Protótipo do ambiente do LOGO3D

3. Resultados Esperados e Conclusões Parciais

Esperamos que o Logo3D auxilie na compreensão de algoritmos, pois ao contrário das linguagens de programação o Logo3D apresenta uma saída gráfica, e não textual, que neste contexto pode ser mais vantajoso observar desenhos decorrentes de algoritmos do que números, fórmulas, relatórios, tabelas, etc.

Outra vantagem da saída gráfica seria a fácil detecção e correção de erros no algoritmo, ou seja, ao contrário do aluno ter que utilizar o tradicional método correção – compilação – execução, a maioria dos comandos do Logo3D corresponde a imagens facilmente identificáveis.

O uso do software como uma ferramenta de ensino ajuda ao individualizar a aprendizagem e agregar estímulos visuais.

Especificamente com o uso da linguagem LOGO, é possível a utilização de uma interface para a construção de conhecimento baseada em desafios cognitivos e na preparação e execução de projetos que tenham relação com o interesse do usuário.

Baseado não na repetição, mas no encontro de soluções para desafios cognitivos, a forma de apresentação dos dados no LOGO3D (visualização) auxiliará na construção das noções espaciais, tais como as de profundidade, largura e comprimento. Este elemento é o grande diferencial desta versão do LOGO das outras ditas somente para crianças.

Assim, será possível elaborar projetos educacionais tridimensionais e ampliar os desafios para os usuários do LOGO.

Referências

- [1] Galembek, Frederico C. M., “LOGO”, São Paulo, Atlas – 1987.
- [2] Malfatti, Silvano e Santos, Selan R., “Implementação do Software Educacional Logo3D – Uma Extensão Tridimensional do LOGO”, Jornada Acadêmica Integrada, Universidade Federal de Santa Maria, outubro de 2000.
- [3] Papert, S., “LOGO: Computadores e Educação”, 3ª edição, Editora Brasiliense, São Paulo, 1985.

Structuring the Space of Interaction Practices in the Segmentation of Medical Images: a First Discussion

Silvia D. Olabarriaga and Marcelo S. Pimenta

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{silvia,mpimenta}@inf.ufrgs.Br

***Abstract.** This paper presents a first discussion on how to structure the space of interaction practices in the segmentation of medical images. We first present an example of application to set a scenario for discussion. Next we summarize some of the interactive aspects related to image segmentation methods. Finally we present some initial ideas about how to improve the design of ISMs with HCI methodology.*

1. Introduction

Segmentation is a step in the analysis of digital images where the object of interest is isolated from the background [Gonzalez and Woods (1993)]. Segmentation supports several analysis tasks such as quantification, inspection and retrieval; in all cases, automatic processing is desirable, but sometimes unattainable due to limitations imposed by image acquisition, abnormalities in the scene, or both. Interactive segmentation methods (ISMs) are adopted in applications with a high demand for accurate results, when the volume of images is reasonable, and when the image dimensionality admits human manipulation.

As a general rule, the goal of ISMs is to combine a human operator and a computer (also named computational part) to obtain an accurate delineation of the object of interest in an efficient manner. The underlying assumption is that the user *knows* the correct delineation and, as such, he/she could guide the computational part to determine it efficiently. This is the case of many applications, in particular in the medical field, where clinical practice and research impose several demands on reliability of results obtained with or by computational tools.

In this short paper we start a discussion on how to structure the space of interaction practices in the segmentation of medical images. We first present an example of application to set a scenario for discussion. Next we present a review of aspects related to interactive segmentation methods. Finally we present some initial ideas about how to improve the design of ISMs with HCI methodology.

2. A simple example of interactive segmentation in medical imaging

As an example of interactive segmentation procedures, consider the following scenario: a pregnant woman is submitted to ultrasound (US) to check the baby's development, which is roughly assessed by the diameter of the baby's head. The doctor manipulates the US probe on the belly, trying to get the best view of the head on the screen, such that its diameter can be most reliably estimated. The image is "frozen" when the doctor is satisfied and then

analysed. The measurement problem at this point could be solved in several ways, ranging from fully manual methods (the doctor uses a ruler on the screen to measure the diameter) to fully automatic methods (the doctor presses a button and the diameter is measured and displayed by the machine).

The first type of solution is not really an option these days - we should be able to do better than that with available digital image analysis technology. The last type of solution, however, is not a realistic option either, since fully automatic methods often fail to reliably isolate the head from the rest of the image, a precondition for measurement. In practice, we will find solutions where the doctor and the machine **interact** to obtain a reliable measurement of the head's diameter.

In the context of this example, user intervention can occur at several moments in the process, such as to initialise the method (the doctor clicks one point inside the baby's head), to check the accuracy of the result produced automatically (the head's contour found by the machine is drawn and the doctor confirms its position before the diameter is estimated), or even to correct the segmentation result manually (the doctor edits the contour produced by the machine before measurement is done).

3. An early review of aspects related to interactive practices in the segmentation of medical images

The interaction strategies adopted by existent methods differ with respect to following three aspects have been discussed by Olabariaga and Smeulders (2001):

- The type of data input by the user during the segmentation process. The main types usually identified are:
 - a) setting parameter values in a continuous or discrete interval;
 - b) pictorial input directly indicating positions on the image grid; and
 - c) choosing among pre-defined menu items or select among pre-computed segmented results.
- The computational consequence of user input, i.e., how the input data are interpreted to feedback the computational part. In the simple case, the input data are directly used as parameters for the computational part. Indirection (that is, interpretation of user input) is mainly aimed at:
 - a) achieving interaction simplicity, typically enabling the user to see only the grey image on the screen and to draw the object with a graphical tool; and
 - b) reducing the amount of user interventions by means of an "intelligent" analysis of user actions and revision of the parameters for the computational part.
- The goal of user intervention in the interactive process. We identify five main purposes (also considered five different user roles, although the user may play more than one role in many situations):
 - a) to judge (accept or reject) the result generated by the computational part;
 - b) to correct the outcome directly using a graphic editor, eventually when and where the computational method fails;

- c) to set parameters for the computational part at two different moments in the computation: to start the method (initialisation) or to provide information that is used to dynamically reconfigure parameter values (steering);
- d) to compose the object delineation from "primitive" results selected among a number of possible delineations generated automatically by the computational part; and
- e) to build a dedicated computational method, choosing the appropriate low-level image processing operations and the corresponding parameters.

These aspects are complementary and found in combination in interactive segmentation methods found in the literature.

4. Discussion

Each aspect presented above has different implications to the ISM's potential usability, determining the accuracy and precision of the result to be achieved and the expected operation efficiency. When designing and evaluating a new interactive method for image segmentation, it is important to carefully choose a combination of concepts, criteria and strategies that can help to improve not only its efficiency, but also its usability.

With respect to the type of data input, we should investigate which are the most adequate interaction styles and widgets for each situation concerning particular set of tasks and users. These are typical problems in HCI: interaction style selection – see Shneiderman (1998), Newman and Lamming (1995) - and interaction object selection – see Bodart and Vanderdonck (1994). The consequence of user input is directly related to low-level dialog design. Sequencing and availability of user actions (user interface syntax) have a large impact on the interaction simplicity – see Newman and Lamming (1995). Likewise, the computational component behavior can be made “intelligent” by anticipating actions or replacing user interventions for interaction efficiency. Finally, the user role in the process is related to high-level dialog design. It indicates the decision level and the degree of freedom the user has with respect to the segmentation goals, determining how much and when the user can interfere in the segmentation process. The examples above represent a limited excursion into a territory that includes many other possible perspectives such as recently explored by Randell and Johnson (2001).

References

- Bodart, F. and Vanderdonck, J. (1994) “On the Problem of Selecting Interaction Objects”. In: *Proc. BCS HCI'94*, August 1994, Cambridge Univ. Press. p. 163-178.
- Gonzalez, R.C. and Woods (1993) R.E. “Image Segmentation”. In: __. *Digital Image Processing*. Addison-Wesley. p.413-481.
- Newman, W.M. and Lamming, M.G. (1995) *Interactive Systems Design*. Addison-Wesley.
- Olabarriaga, S. and Smeulders, A.W.M. (2001) “Interaction in the Segmentation of Medical Images: a Survey”. *Medical Image Analysis*, v.5. p.127-142.
- Randell, R. and Johnson, C. (2001) “A Comparison of Methods for Analysing Human-Computer Interaction Failure in the Medical Domain”. Tech. Report. Univ. of Glasgow. <http://www.dcs.gla.ac.uk/~rebecca/comparison.pdf>
- Shneiderman, B. (1998) *Designing the User Interface*, Addison-Wesley.

Sugestões para a Criação de Documentos *Web* - Visando a Usabilidade

Eliane da S. A. Diniz^{1,2}, Cristiane O. Vieira²

¹Pós-Graduação em Computação, Instituto de Informática, UFRGS

²Instituto de Física e Matemática, UFPEL

diniz@inf.ufrgs.br, chris@ufpel.tche.br

Resumo. *Nossos estudos se concentram no projeto de interfaces com usuário na World Wide Web, onde encontramos vários problemas decorrentes do mau uso dos elementos de interface na criação dos hipertextos, o que dificulta a navegação. Sem limitar a criatividade do designer, elaboramos uma série de sugestões concisa, prática e de bom embasamento teórico, que permita a geração de documentos Web com usabilidade e de fácil confecção.*

Palavras-chave: *Web, Websites, Usabilidade, Interface com o Usuário, Design e Diretrizes.*

1. Introdução

Através da observação da grande variedade de estilos utilizados nas páginas da *Web*, percebemos que existem poucos critérios para a sua construção, o que torna a navegação confusa, maçante e exaustiva [3], prejudicando o usuário e também o projetista - que muitas vezes precisa "adivinhar" como projetar uma interface para a Rede. Por esse motivo, algumas tentativas de padronização de *Web sites* têm surgido, através de diretrizes para a sua construção, porém trata-se ainda de um processo embrionário com normas conflitivas e pouco abrangentes. A partir dessas premissas, percebemos a necessidade de avaliar essas diretrizes comparando-as com as já estabelecidas normas para construção de interface, e criar uma série de sugestões que facilitem a confecção de *Web sites* com melhor aparência, funcionalidade e principalmente usabilidade.

2. Elementos para o Projeto de Interfaces Amigáveis

Alguns autores {[2] e [18]} relacionam vários fatores importantes na construção de interface para software, entre eles: projeto de interface; tipo de usuário e de tarefa; aspectos econômicos, cognitivos e ergonômicos. Além desses fatores, para que o processo de criação da interface se desenvolva de maneira eficaz e organizada é importante utilizar métodos de projeto de Interface com o Usuário - IU, tais como: avaliações de heurísticas, design participativo, acompanhamento cognitivo, análise de tarefas, prototipação rápida e diretrizes para projeto.

Embora neste trabalho nos propomos somente a reunir um conjunto de diretrizes de projeto de IU que se aplique ao ambiente *Web* (seção 4), devemos salientar que o designer não deverá optar por apenas um dos métodos vistos acima, como base para todo o processo de criação de IU, a fim de conseguir uma interface sensível ao contexto, ao usuário e a tarefa.

3. Sugestões para a Criação de Documentos na Web

As sugestões para construção de páginas *Web*, serão reunidas em quatro categorias: Organização da Informação, Elementos Gráficos, Comunicação e Navegação e foram baseadas em NASA ([1]), Nielsen ([2] até [13]), e Shneiderman [14]. Enfatizamos que essas podem ser modificadas, levando-se em conta as alterações constantes dos elementos disponíveis para a construção de *Web sites*, bem como a evolução dos *browsers*. A seguir serão apresentadas como ilustração algumas das sugestões dentre as obtidas.

3.1 Organização da Informação

3.2. Ao usar *frames*, limite-se a 1 ou 2 divisões da página, de maneira a não retalhar a tela.

4.2. Apresente as informações de forma hierárquica, com os grupos mais genéricos na página principal, criando uma estrutura de tópicos consistente e eficiente.

4.6. Separe os grupos lógicos de informações por divisores (linhas, espaços ou cores), facilitando o entendimento.

5.3. Evite o uso de barras de rolagem, experimente subdividir o assunto.

6.1. Permita a visualização do *site* em mais de um idioma, tornando-o acessível a usuários de diferentes países.

3.2. Elementos Gráficos

8.2. Utilize no máximo 4 variações no tamanho das fontes, causando uma boa visualização dos tópicos realmente importantes.

10. Use inversão de cores no lugar de inversão de vídeo (*reverse video*) quando precisar direcionar a atenção para um certo aspecto do *site*.

17.1. Use figuras para ilustrar seus comentários.

18.1. Quando for utilizar texturas ou panos de fundo, evite aquelas com logomarcas ou palavras, que dificultam a leitura do texto colocado sobre elas.

19.7. Se for usar *banners*, limite-se a um ou dois por página, para não prejudicar a navegação e o tempo de *download* do *site*.

3.3. Comunicação

24.1 Use títulos concisos e significativos, exaltando a boa comunicação do sistema com o usuário.

28. Utilize os *hints* e a barra de *status* para informar o usuário sobre ações relevantes a tarefa em andamento, salientando a confiança de seu *site*.

3.4. Navegação

43.3. Considere um pequeno *link* no fim da página para a visualização de informações antigas, de maneira a acomodar as preferências de usuários fiéis.

43.4. Use logomarcas com *links* para a *home page* do *site* em todas as páginas secundárias do mesmo, exceto a página inicial, prevenindo acessos desnecessários ao servidor.

46.6. Opte por permitir o *download* (FTP) de arquivos com versões antigas do *site*, para que usuários possam compará-lo com as novas versões.

4. Exemplos de páginas com referências as sugestões

Mostraremos algumas páginas *Web* onde serão observados os elementos de interface mencionados nas sugestões relacionadas acima, caracterizando-os como:

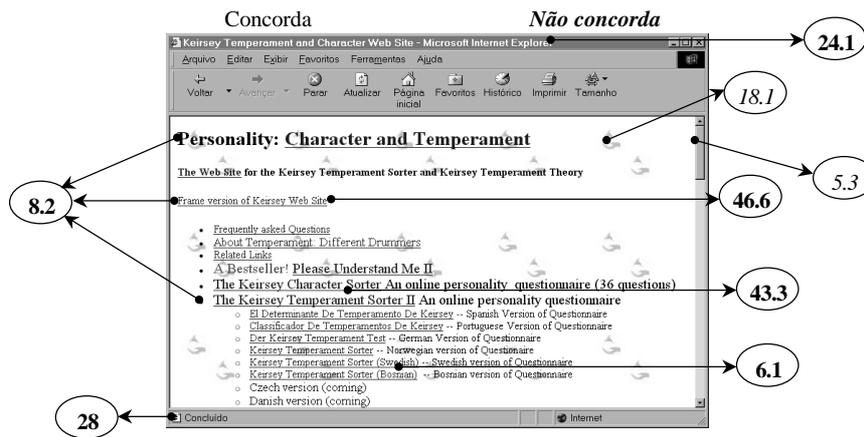


FIGURA 1 – Análise de fontes, texturas, idioma, conteúdo, título, barra de rolagem, barra de status e disponibilidade de versões diferentes em um site.



FIGURA 2 – Análise de banners, figuras, realce, frames, organização da informação e atalhos em site.

5. Conclusões

A construção de interface depende na verdade do bom senso de cada projetista, e da observação cuidadosa das peculiaridades existentes em cada usuário e tarefa envolvidos. Com o desenvolvimento de ferramentas que auxiliam leigos a construírem documentos

Web, encontramos o erro mais cometido na elaboração de IU, o excesso. A maioria dos projetistas não consegue limitar o uso de elementos conhecidos como tecnologia de ponta.

A partir daí, procuramos construir um conjunto de sugestões para a elaboração de *Web sites*, de redação clara e simples, que auxilie o projetista a ter como resultado uma IU com usabilidade. Devemos informar que temos obtido resultados satisfatórios na utilização dessas sugestões durante o processo de construção de *Web sites*, e que essas são genéricas aos *sites* em geral, não se aplicando a IU utilizadas em nichos de interfaces especializadas.

Bibliografia

- [1] NASA/GODDARD SPACE FLIGHT CENTER; **User Interface Guidelines**. Jan./1996
http://groucho.gsfc.nasa.gov/Code_520/Code_522/Documents/UG_96/newfrontmatter
- [2] NIELSEN, J.; *Hypertext and Hypermedia*, Academic Press, 1990.
- [3] NIELSEN, J.; MACK, R.L.; *Usability Inspections Methods*. New York: John Wiley, 1994. 413p.
- [4] NIELSEN, J.; *Guidelines for Multimedia on the Web*. www.useit.com/alertbox/, 1995.
- [5] NIELSEN, J.; *Be Succinct! (Writing for the Web)*. www.useit.com/alertbox/, 1997.
- [6] NIELSEN, J.; *The Tyranny of the Page: Continued Lack of Decent Navigation Support in Version 4 Browsers*. www.useit.com/alertbox/, 1997.
- [7] NIELSEN, J.; *Web Pages Must Live Forever*. www.useit.com/alertbox/, 1998.
- [8] NIELSEN, J.; *When Bad Design Elements Become the Standard*. www.useit.com/alertbox/, 1999.
- [9] NIELSEN, J.; *“Top Ten Mistakes” Revisited Three Years Later*. www.useit.com/alertbox/, 1999.
- [10] NIELSEN, J.; *The Top Ten New Mistakes of Web Design*. www.useit.com/alertbox/, 1999.
- [11] NIELSEN, J.; *Ten Good Deeds in Web Design*. www.useit.com/alertbox/, 1999.
- [12] NIELSEN, J.; *Prioritize: Good Contents Bubbles to the Top*. www.useit.com/alertbox/, 1999.
- [13] NIELSEN, J.; *Flash: 99% Bad*. www.useit.com/alertbox/, 2000.
- [14] SHNEIDERMAN, B.; *Designing Information-Abundant Websites: Issues and Recommendations*. 1997.

Superagui Roleplaying Game: Proposta de uma ferramenta na sensibilização para educação ambiental

**Ana Cláudia Bevacqua, Juliana Schwartz, Hilton José da Silva Azevedo,
Líbia Patricia Peralta Agudelo, Marcelo Stein de Lima Sousa**

Programa de Pós-Graduação em Tecnologia - Centro Federal de Educação
Tecnológica do Paraná (CEFET - PR)
Av. Sete de Setembro, 3165 - Bloco D - Rebouças
80230-901 - Curitiba - PR - Brasil

{bevacqua, schwartz, hilton, pperalta, msteinas}@ppgte.cefetpr.br

***Abstract.** This work presents a RolePlaying Game (RPG) as a software to be used as a sensitization tool in a distance learning course. This software's objective is to help understand and solve social and environmental conflicts that occur in conservation units, like the Superagui National Park (PNS), located at the coast of the Parana State, southern Brazil. Since many governmental and non-governmental sector are involved in managing the park. The game will encourage them to cooperate and better understand each others activities and their impact on the local communities.*

***Resumo:** Este trabalho apresenta um RolePlaying Game — RPG (Jogo de Representação) enquanto ambiente computacional a ser utilizado como ferramenta de sensibilização de um curso de ensino à distância, voltado para a educação ambiental. Esta ferramenta tem como objetivo subsidiar o entendimento e resolução de conflitos sócio-ambientais que ocorrem no entorno de unidades de conservação, no caso o Parque Nacional do Superagui (PNS), localizado no litoral norte do Paraná, já que os vários atores envolvidos na gestão do parque viriam a desempenhar papéis que facilitariam um melhor entendimento de sua atuação sobre a comunidade local.*

Introdução

O Parque Nacional do Superagui é um exemplo de situação crítica que ocorre em unidades de preservação, pois existem conflitos sócio-ambientais tais como o acesso aos recursos naturais, a gestão e o uso do entorno do parque pelas comunidades locais. Há o problema da falta de comunicação entre os interessados e também determinantes sociais e ambientais que afetam as decisões sobre o seu manejo. Um Projeto viabilizado através de convênio entre a Curadoria do Patrimônio Natural Tombado (CPNT) da Secretaria Estadual da Cultura do Paraná (SEEC) e o Centro Federal de Educação Tecnológica do Paraná (CEFET-PR), visa desenvolver subsídios tecnológicos que facilitem o diálogo dos atores envolvidos no PNS (tombado em 1985) que integra diversas variáveis sociais, ambientais e econômicas. Devemos observar que esta situação é comum a outros contextos tanto urbanos quanto rurais, mas toma proporções ainda mais críticas no entorno de unidades de preservação ou bens tombados.

Como alternativa de sensibilização na questão de preservação ambiental e em relação aos conflitos sócio-ambientais do PNS, sugerimos a criação e a implementação de um curso de Alfabetização Ambiental à distância direcionado inicialmente à equipe técnica da CPNT. Um dos componentes deste curso é a utilização de um RPG como ambiente de aprendizagem, levando em consideração a Teoria da Atividade, como desenvolvida por Yrjö Engeström (1987).

Superagüi RPG

RPG é um jogo onde cada participante faz o papel de uma personagem, tomando parte de uma aventura imaginária. O tipo de aventura é definido por um árbitro chamado Mestre, é ele quem determina o cenário e faz o papel dos “personagens incidentais” que os outros personagens encontrarão no decorrer do jogo [Jackson 1994]. Uma característica importante do RPG é que ele não precisa ser competitivo. Na maioria das situações o grupo terá, ou não, conseguido realizar seu objetivo conjuntamente, dependendo do quanto eles colaborarem entre si. Outra característica é que ele é um jogo muito peculiar, pois não há vencedores, é no desenrolar das histórias que surgem as derrotas e as vitórias, altos e baixos, que somados ao fim, garantem ao participante a satisfação de ter atuado como um viajante dos caminhos que a imaginação da equipe permitiu trilhar.

A escolha de um jogo em formato RPG se deu em função do ciclo expansivo, pois na Teoria da Atividade dois processos básicos são considerados como contínuos e interdependentes no desenvolvimento das atividades humanas: a internalização e a externalização. O primeiro está relacionado com a reprodução da cultura: o ser humano internaliza conhecimentos, conceitos, valores e significados reproduzindo-os em suas relações sociais. O segundo está ligado à capacidade criativa do ser humano, através da qual é possível transformar a realidade vivida. Logo, um ambiente criado com base nos atores da comunidade do Superagüi permite com que a reprodução da cultura local (conceitos, valores, crenças, significados), mostrada através de seus personagens, crie condições para internalização das estruturas sociais que fazem parte deste contexto. A partir do momento em que esses conceitos são internalizados aumenta a possibilidade de transformação e de criação de ferramentas mediadoras que possam influenciar o desenvolvimento das organizações sociais. Um outro fator que faz com que o RPG seja uma opção favorável neste caso é que em sendo uma atividade lúdica, ele vem ao encontro de estruturas básicas da psique humana (que dizem respeito a motivação), além do que aprender é uma atividade humana básica, que acontece a todo momento, é fundamentalmente social, ou seja, através do aprendizado criamos nossa identidade dentro da comunidade de prática [Wenger 1998] à qual pertencemos ou queremos vir a pertencer, é através do processo de “troca” com o Outro que construímos nossos valores e nossas representações psíquicas da realidade concreta.

Neste caso, o jogo tem como cenário o próprio Parque do Superagüi, e é composto de cinco personagens, que representam algumas das comunidades que fazem parte do contexto social, cada uma com características e atividades próprias. O jogo será dividido em etapas, cada uma delas composta de obstáculos e/ou problemas apresentados pelo Mestre do Jogo. Cada personagem terá sua própria ambientação onde será possível interagir e negociar com outros personagens e com o Mestre do Jogo.

O objetivo é que os diferentes atores entendam seu papel e possam se colocar na posição de negociar com os outros envolvidos nessa atividade, pois na prática, o que acontece atualmente, é que tanto técnicos dos órgãos de gestão como os moradores locais atuam de forma isolada, o que dificulta a negociação e resolução dos conflitos.

Mesmo cada participante assumindo somente um papel específico no jogo, ao ter que negociar com os outros, poderá entrar em contato com pontos de vista e necessidades até então desconhecidas por ele. Nesse caso espera-se que o RPG dê uma visão integrada do contexto do PNS. Como forma de avaliar essa proposta haveria um monitoramento junto aos atores envolvidos, para verificar se o RPG teria ajudado a identificar com mais clareza os conflitos sócio-ambientais existentes e se isto teria contribuído para facilitar o diálogo e negociação da compreensão global do problema.

Considerações Finais

Ao pensarmos no RPG como uma ferramenta de sensibilização, tentamos reproduzir, em um ambiente educacional, um contexto que muitas vezes se torna distante do dia a dia do aprendiz.

Essa ferramenta nos traz o desafio de desenvolver uma interface de RPG voltada para o trabalho colaborativo, onde um de seus objetivos é fixar conceitos desenvolvidos no curso de alfabetização ambiental a partir de visões próprias das comunidades em que esses participantes irão atuar. Esse ambiente inclui elementos que vem dar suporte ao trabalho em grupo tais como: um mediador no papel de Mestre do jogo que conduz e orienta as ações dos outros jogadores; a inserção de valores sociais considerados tabus no próprio roteiro do RPG e o design do jogo o mais próximo possível do contexto real da atividade dos personagens, já que seu protótipo está sendo feito com base em observação direta no PNS.

Bibliografia

- Engeström, Yrjö. Learning by Expanding: An Activity-Theoretical Approach to Developmental Research. Orienta Konsultif Oy, 1987.
- Jackson, Steve. GURPS: Generic Universal Roleplaying System – Módulo Básico. Devir Editora, 1994.
- Wenger, Etienne. Communities of Practice: Learning, Meaning and Identity. Cambridge University Press, 1998.

Um Agente Inteligente Baseado na Engenharia Semiótica Inserido em um Ambiente de Aprendizado à Distância

João Jung¹, Patrícia Jaques¹, Adja Andrade^{2,3}, Rafael Bordini¹, Rosa Vicari¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²PGIE- Programa de Pós Graduação em Informática na Educação (UFRGS)

³FACIN- Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS).

{jjung, pjaques, bordini, rosa}@inf.ufrgs.br; adja@inf.pucrs.br

Resumo: Este artigo propõe a concepção de um agente semiótico como parte de um ambiente de Educação à Distância concebido sob a visão de aprendizagem situada, isto é, na concepção de cognição como uma prática social. Para isso, utiliza-se da Engenharia Semiótica cuja abordagem apresenta uma perspectiva para Interação Humano-Computador na qual o sistema computacional é um artefato de metacomunicação para projeto de interfaces de usuário.

1. Introdução

O nosso trabalho visa a concepção de um agente semiótico inserido em um ambiente educacional baseado na teoria sócio-interacionista segundo Vygotsky (1998), Levy (1999) e Freire e Fagundes (1995). Este ambiente foi projetado como um sistema multiagente composto por agentes artificiais e agentes humanos. Um dos agentes artificiais deste ambiente, o agente semiótico, responsável pela utilização de signos, símbolos, conceitos e linguagem, fará uso da Engenharia Semiótica através de modelos e formalismos definidos por Leite (1998) e Leite e de Souza (1999). Estes formalismos auxiliam o processo de *design* da interface feito pelo agente semiótico e tem o objetivo de facilitar a comunicação e/ou a aquisição de conhecimento por parte dos usuários.

2. Fundamentação Teórica

A abordagem da Engenharia Semiótica, segundo de Souza (1993), considera o sistema computacional, mais especificamente a interface, como um artefato de *metacomunicação*, cujo conteúdo deve ser o modelo de usabilidade abrangendo a interação e a funcionalidade do sistema, através do processo de *design* como uma atividade de produção de signo.

A semiótica é definida por Eco (1976) como a disciplina responsável por estudar os signos, sistemas de signos, significação, comunicação e todos os processos culturais.

Para dar embasamento ao modelo coletivo de Educação à Distância (EAD), este projeto, definido por Andrade *et al.* (2001), utiliza-se de um dos conceitos importantes da teoria de Vygotsky (1998), onde o funcionamento psicológico está baseado nas relações sociais entre o indivíduo e o meio em um processo histórico e que a relação homem-meio é uma relação mediada por sistemas simbólicos, através de instrumentos e signos. Para Vygotsky, os signos são estímulos artificiais com a finalidade de auxílio mnemônico; eles funcionam como meio de adaptação, dirigido para o controle do próprio indivíduo. O signo é

orientado internamente. Já a função de um instrumento é servir como um condutor da influência humana sobre o objeto da atividade; este é orientado externamente. Ambos têm em comum, a função de mediação.

Sistema Multiagente (SMA) é uma área da Inteligência Artificial Distribuída em que sistemas são elaborados a partir de componentes de software chamados agentes. Cada agente possui seus próprios objetivos e são autônomos no sentido de que somente realizam tarefas por decisão própria. Em um SMA, os agentes devem interagir para coordenar suas ações no sentido de atingirem objetivos comuns, que são frequentemente impossíveis de serem atingidos com as capacidades individuais destes agentes. Um apanhado geral da área é dado por Jennings *et al.* (1998).

A tecnologia de agentes pode trazer significativo suporte à aprendizagem na Web. As características de autonomia, colaboração e aprendizagem podem auxiliar na construção de um modelo de aprendiz e auxiliar na interação entre os alunos, incentivando o seu desenvolvimento sócio-cognitivo.

3. Definição do Projeto: Agente Semiótico

O modelo pedagógico proposto nesta pesquisa, segundo Andrade *et al.* (2001), está fundamentado na busca por uma forma colaborativa de aprendizagem que se efetive através da interação social. O sistema é formado por agentes humanos (aprendizes e tutores) e por cinco classes de agentes artificiais: *agente diagnóstico* que visa implementar, da melhor maneira possível, o conceito de Zona de Desenvolvimento Proximal proposto por Vygotsky, visando transformar habilidades potenciais em habilidades reais e buscando expandir a capacidade de desenvolvimento sócio-cognitivo do aluno; *agente mediador* que auxilia no processo de internalização do aluno decorrente do contato com o ambiente social de EAD; *agente social* que deve estabelecer a integração da sociedade e construir os modelos de grupos de aprendizes; *agente colaborativo* que é responsável por mediar/monitorar a interação síncrona entre os alunos; e **agente semiótico**, que é o objetivo desta pesquisa, responsável pela utilização de símbolos (signos e instrumentos), conceitos e linguagem que serão apresentados como material instrucional para o aluno.

Para que o agente mediador possa cumprir o seu papel, é necessária a intervenção de estímulos externos na forma de instrumentos e signos. Estes elementos são introduzidos pelo agente semiótico para auxiliar na atividade cognitiva do aluno a fim de solucionar um dado problema. Para tanto, o agente semiótico poderá utilizar vários signos, dentre eles: o desenho, a escrita, o sistema de números, recursos multimídia, hipertextos e animações para propiciar táticas de ensino adequadas aos aprendizes.

4. Agente Semiótico e Engenharia Semiótica

Tomando como base a abordagem da Engenharia Semiótica, consideramos o termo *designer* como sendo o nosso **agente semiótico**, que desempenha este papel. O agente semiótico decide quais os signos que melhor se enquadram a um certo aprendiz em um certo momento, dependendo da tática de ensino especificada pelo agente diagnóstico e repassado pelo agente mediador.

Para isso, é importante termos uma modelagem de quais signos serão utilizados e como apresentá-los ao usuário. Assim, nos baseamos nos trabalhos propostos por Leite e Souza (1999) e Leite (1998) para a especificação das regras de decisão do agente

semiótico. Estas regras são utilizadas na construção de *design* das interfaces de usuário, utilizando-se da Linguagem de Especificação da Mensagem do *Designer* que tem como objetivo apoiar a formulação de mensagens sobre o modelo de usabilidade. Mensagens estas que podem ser: sobre estados de signos do domínio, sobre funções da aplicação, sobre a estrutura sintática dos comandos, sobre interações básicas, de metacomunicação de assistência a tarefas, de metacomunicação para apresentação e controle da leitura da mensagem, e de metacomunicação direta. Um exemplo de mensagem segue abaixo:

```
Command-Message Exercício for Application-Function Exercícios
  Join { Sequence { View Information-of Enunciado do exercício
    Repeat { View Information-of Itens/Opções do exercício }
    Join { Enter Information-of Resposta do aluno
      Activate Show Command-Message Resolução/gabarito } }
  Select { Activate Start Application-Function Exercícios
    Activate Waive Application-Function Exercícios } }
```

5. Conclusões

Através da abordagem da Engenharia Semiótica, podemos explorar o amplo potencial metacomunicativo existente no projeto de interfaces dos sistemas computacionais. Assim, através de uma perspectiva original e inovadora, podemos simplificar e interpretar problemas tradicionais no que diz respeito ao processo de *design* de interfaces e interação usuário-sistema, utilizando-se da abordagem semiótica para este projeto de aprendizagem fundamentado na teoria sócio-interacionista de Vygotsky.

Uma boa interface de usuário e com qualidade é ainda mais crucial em sistemas de ensino, pois além de realizarem o papel de mediador e facilitador entre aluno e conteúdo, elas devem estimular o aluno no seu aprendizado. Neste sentido, o trabalho apresentado neste artigo tem por objetivo permitir um melhor desempenho nos aspectos de usabilidade, o que implica em interfaces com melhor interação e funcionalidade.

6. Referências Bibliográficas

- Andrade, A.; Jaques, P.; Vicari, R.; Bordini, R.; Jung, J. (2001) A Computational Model of Distance Learning Based on Vygotsky's Socio-Cultural Approach. *Proceedings of the MABLE Workshop*. X International Conference on Artificial Intelligence in Education. Antonio, Texas, May 19-23 2001.
- de Souza, C. S. (1993) The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies*, 39. Academic Press. pp.753-773.
- Eco, U. (1976) *A Theory of Semiotics*. Bloomington. Indiana University Press.
- Freire, P.; Fagundes, A. (1995) *Por uma Pedagogia da Pergunta*. Ed. Paz e Terra, Rio de Janeiro.
- Jennings, N. R.; Sycara, K.; Wooldrige, M. (1998) A Roadmap of Agent Research and Development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1). Kluwer Academic Publishers, Boston.
- Leite, J. C. (1998) *Modelos e Formalismos para a Engenharia Semiótica de Interfaces de Usuário*. Tese de Doutorado. Departamento de Informática. PUC-Rio.
- Leite, J. C.; de Souza, C. S. (1999) Uma Linguagem de Especificação para a Engenharia Semiótica de Interfaces de Usuário. *Proceedings of the IHC'99*. Campinas, SP, Brazil, October.
- Levy, P. (1999) *Inteligência Coletiva. Por uma antropologia do ciberespaço*. Edições Loyola, São Paulo.
- Vygotsky, L. S. (1998) *A Formação Social da Mente: o Desenvolvimento dos Processos Psicológicos Superiores*. Editora Martins Fontes, São Paulo.

Validação de Recomendações Propostas para Projeto de Interfaces de Aplicações Institucionais de Aprendizagem

Marilda S. Chiaramonte e Cora H. F. Pinto Ribeiro

Universidade Federal do Rio Grande do Sul
Instituto de Informática – PPGCC
Av. Bento Gonçalves, 9500 - Bairro Agronomia
91.501-970 - Porto Alegre - RS – BRASIL
Fax +55(51)3336 5576
<http://www.inf.ufrgs.br>

marildac@italnet.com.br, cora@inf.ufrgs.br

***Abstract.** This Poster presents the results of an experimental test performed for basic interface design guidelines validation. The goal is to help designers in the development of applications with interfaces that positively support the learning process, by using criteria corroborated by experimental validation. This work also presents the study description, the collected data, and the results of the investigation.*

***Keywords:** Graphic design, interface design, navigational context*

1. Introdução

As interfaces gráficas utilizadas em diferentes aplicações disponíveis na *Web* deveriam ser adequadas a cada contexto. A validação prévia dos critérios adotados, específicos, como referentes aos aspectos gráficos e o contexto de navegação, possibilitaria uma maior adequação aos objetivos da aplicação e ao perfil do público alvo. Profissionais provenientes de diferentes áreas do conhecimento, como especialistas em informática, artistas gráficos, professores e fisiologistas, valorizam aspectos distintos de uma interface, que nem sempre são compatíveis (como na utilização de cores, com um efeito artístico de impacto mas inadequadas às características fisiológicas da visão). Outra observação refere-se às diferenças associadas a características específicas de um ambiente de aplicação (como o tempo de exposição de um usuário à tela em uma aplicação de ensino ou de reservas de passagens) ou dos diferentes perfis de usuário (diferenças de visão dos usuários de aplicações voltadas para adolescentes ou para pessoas da terceira idade [4]), as quais usualmente não são consideradas. De maneira geral, as recomendações encontradas na literatura [1, 3, 7, 8 e 9] ou observadas na prática são contraditórias. Observamos também que, embora a validação experimental seja defendida por autores reconhecidos, como Nielsen [5,6], esta não é utilizada como base das propostas existentes.

O objetivo deste trabalho é de identificar, a partir da validação experimental, critérios básicos a serem adotados no projeto gráfico de interfaces voltadas para aplicações de aprendizagem. As premissas iniciais são que o tempo de exposição à aplicação pode ser longo e que os usuários podem apresentar características bastante distintas de idade, nível de conhecimento, experiência prévia no uso de computadores, características cognitivas e qualidade de visão.

Na etapa referente ao contexto de navegação (Tabela 2), buscou-se estabelecer uma relação entre o perfil visual ou verbal¹ do usuário e o contexto de navegação selecionado.

O instrumento investigou a preferência por elementos de navegação textuais ou icônicos e pela posição destes na tela da aplicação. Os resultados apresentados demonstraram, nos dois grupos, uma preferência para a navegação através de componentes textuais. Em relação à posição do elemento navegacional na tela da aplicação, os resultados mostraram a preferência, em ambos os grupos e com bastante significância, pela localização do elemento de navegação à direita, seguida pela posição à esquerda.

Tabela 2: Preferências em relação ao contexto de navegação (tipo e posição do elemento)

Navegação	Figura-texto								Navegação	localização							
	Total		Visual		Verbal		não identif.			Total		Visual		Verbal		não identif.	
	No	%	No	%	No	%	No	%		No	%	No	%	No	%	No	%
texto	168	42%	120	41%	32	54%	16	33%	direito	161	40%	124	42%	24	41%	13	27%
figura com texto	85	21%	64	22%	14	24%	7	14%	esquerdo	80	20%	55	19%	14	24%	11	22%
figura	9	2%	7	2%	0	0%	2	4%	em cima	13	3%	7	2%	4	7%	2	4%
sem preferência	72	18%	60	20%	7	12%	5	10%	embaixo	4	1%	2	1%	2	3%	0	0%
não navegaram	69	17%	44	15%	6	10%	19	39%	sem preferência	76	19%	63	21%	9	15%	4	8%
									não navegaram	69	17%	44	15%	6	10%	19	39%

4. Conclusão

Este trabalho contribuiu através de um estudo experimental, buscando a identificação dos fundamentos para o projeto de interfaces de aplicações de aprendizagem, a partir da validação experimental. Embora os aspectos analisados sejam limitados, os resultados mostram a importância da validação experimental e o perigo da utilização de critérios que não sejam validados a partir deste tipo de investigação.

Alguns resultados obtidos no experimento prático são compatíveis com a fundamentação teórica ou recomendações de manuais e guias disponíveis (como na utilização de cores de fundo de tela), enquanto que outros apontam uma contradição entre os resultados práticos e a teoria recomendada (como nos fontes utilizados para textos). Em relação às características de uma interface para as quais os usuários não demonstraram preferência (como na distribuição de conteúdos em uma única ou em diversas telas), cabe salientar que o uso de diferentes telas pode resultar em um maior tempo de espera, uma vez que requer que cada uma seja trazida individualmente. Os resultados da pesquisa apontaram para uma tendência significativa no uso de elementos textuais para a navegação. Assim, recomenda-se que os projetistas de aplicações de ensino componham os recursos de navegação, apresentando-os, preferencialmente, nesta forma. Em relação ao posicionamento na tela, os elementos de navegação devem estar preferencialmente dispostos no lado direito da tela.

¹ Felder [2] caracteriza os diferentes estilos individuais de aprendizagem como: ativos (discutem e elaboram) e reflexivos (pensam a respeito); sensitivos (aprendem através de metáfora) e intuitivos (relacionam conceitos); seqüenciais (aprendem por etapas) e globais (efetua associações aleatórias); visuais (aprendem com imagens) e verbais (aprendem com palavras ou textos).

Este mesmo instrumento será aplicado em outros ambientes de aprendizagem, para uma revalidação dos resultados obtidos e posterior criação de um modelo de recomendações aplicáveis a estes ambientes.

Referências Bibliográficas

- [1] IBM Web Guidelines – Complete Set, 1999; Disponível em <http://www.ibm.com/ibm/hci/guidelines/web/print.html>
- [2] Felder, R. M.; Soloman, B.A.; *Learning Styles and Strategies*; North Carolina State University; 1999. Disponível em <http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.html>
- [3] Lynch, J. P.; Horton, S.; *Web Style Guide*. Center for Advanced Instructional Media (CAIM), Yale University; 1997. Disponível em <http://www.info.med.yale.edu/caim/manual>
- [4] Nemetz, F.; Lima, J. V. de; Borges, A. C.; *Authoring a Literary Hypermedia Encyclopedia SD-ROM Using Hypermedia Modeling Technique*. Revista de Informática Teórica e Aplicada, Brasil, v.5, n.2, p.55-63, 1998.
- [5] Nielsen, J.; *Are Users stupid?*; Jakob Nielsen's Alertbox; 2001. Disponível em <http://www.useit.com/alertbox/20010204.html>
- [6] Nielsen, J.; *Is Navigation Useful?*; Jakob Nielsen's Alertbox; 2000. Disponível em <http://www.useit.com/alertbox/20000109.html>
- [7] Parizotto, R.; *Elaboração de um Guia de Estilos para serviços de Informação em Ciência e Tecnologia via Web*. Dissertação de Mestrado, UFSC, 1997.
- [8] Valiati, E. R. A.; *Guia de Recomendações para o desenvolvimento de interfaces com usabilidade em Softwares Educacionais do tipo hipertexto/hipermídia Informativo*. UFRGS, 1999.
- [9] Winckler, M. A. A.; *Proposta de uma Metodologia para Avaliação de Usabilidade de Interfaces WWW*. Dissertação de Mestrado, UFRGS, 1999.

Índice por Autor

Artigos

Adéle Malta Pontes.....	207
Afonso Inácio Orth.....	207
Ana Carolina Bertoletti.....	108
Ana Maria Nicolaci-da-Costa.....	120
Ana R. Carballo.....	183
Antônio C. da Rocha Costa.....	108
Antônio Marcos Endler.....	96
Andréa dos Santos Rodrigues.....	14
Carla Faria Leitão.....	120
Carlos Proal.....	183
Clarissa Maria de Almeida Barbosa.....	25
Clarisse Sieckenius de Souza.....	25, 36, 229, 242
Claudia Bauzer Medeiros.....	157
Cláudia Dias.....	73
Christelle Farenc.....	48
Cristiano L. Soares.....	217
Evandro Franzen.....	132
Dámaris Pérez.....	183
Daniela Romão-Dias.....	120
Fernanda Maria Pereira Freire.....	84
Flávio Sanson Fogliatto.....	96
Gilda Dahis.....	36
Guido Lemos de Souza Filho.....	14
Heloísa Vieira da Rocha.....	84, 169
Jair Leite Cavalcanti.....	1
Janne Yukiko Y. Oeiras.....	84
Jiani Cordeiro Cardoso.....	195
J. Alfredo Sánchez.....	183
João Batista Oliveira.....	195
José Antônio Borges.....	14
Júnia C. Anacleto Silva.....	217
Juliana Salles.....	61
Juliano Schimiguel.....	157
Kamran Sedig.....	229
Lia Buarque Macedo de Guimarães.....	96
Luciana Alvim Santos Romani.....	84, 169
Manoel Gomes de Mendonça Neto.....	256
Marcelo Pimenta.....	48, 132
Márcia Cristina Moraes.....	108
Márcio Oliveira Almeida.....	256
Marco Winckler.....	48
Maria Cecília C. Baranauskas.....	61, 157

Milene Selbach Silveira.....	145
Patrícia A. Jaques.....	132
Philippe Palanque.....	48
Raquel Oliveira Prates.....	25
Roberto S. Bigonha.....	61
Rosa Vicari.....	132
Rosângela D. Penteado.....	217
Sérgio Roberto P. da Silva.....	242
Simone Diniz Junqueira Barbosa.....	36, 145
Tânia Kist.....	132
Tatiana. A. Tavares.....	1

Posters

Adja Andrade.....	307
Ana Cláudia Bevacqua.....	304
André Junqueira Xavier.....	275
André Raabe.....	275
Anna Lúcia dos Santos Vieira e Silva.....	288
Catharine Ferreira Bach.....	290
Cora H. F. Pinto Ribeiro.....	310
Cristiane O. Vieira.....	300
Deise Juliana Francisco.....	294
Eliane da S. A Diniz.....	300
Elizabeth Furtado.....	284
Kelly Cristina Goedert.....	278
Hilton José da Silva Azevedo.....	304
João Jung.....	307
Juliana Schwartz.....	304
Líbia Patricia Peralta Agudelo.....	304
Marcelo Soares Pimenta.....	297
Marcelo Stein de Lima Sousa.....	304
Márcia Barros de Sales.....	275
Maria Augusta Silveira Netto Nunes.....	294
Maria Esther Russo Lima.....	271
Marilda S. Chiaramonte.....	310
Mario Alberto Moreno Rocha.....	281
Patrícia Jaques.....	307
Rafael Bordini.....	307
Raquel Oliveira Prates.....	290
Régis Simão.....	284
Rosa Maria Videira de Figueiredo.....	290
Rosa Vicari.....	307
Silvano Maneck Malfatti.....	294
Selan Rodrigues dos Santos.....	294
Sílvia D. Olabarriga.....	297
Vera R. N. Schuhmacher.....	278
Walter de Abreu Cybis.....	271, 278