Anais do
III Simpósio Brasileiro de Sistemas Colaborativos
SBSC 2006

Natal, RN - Brasil

20 a 22 de Novembro de 2006

Promoção

Sociedade Brasileira de Computação

Comissão Especial de Sistemas Colaborativos

Realização

Universidade Federal do Rio de Janeiro

Pontifícia Universidade Católica do Rio de Janeiro

Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto / USP

Universidade Federal do Rio Grande do Norte

Editores

Alberto Barbosa Raposo

Flávia Maria Santoro

© Sociedade Brasileira de Computação
Anais do III Simpósio Brasileiro de Sistemas Colaborativos – SBSC 2006

2006 – Direitos desta edição reservados à Sociedade Brasileira de Computação
ISBN 85-7669-097-7

Editores
Alberto Barbosa Raposo
Flávia Maria Santoro

Para cópias adicionais destes anais, enviar pedido para:
Sociedade Brasileira de Computação
Av. Bento Gonçalves, 9500 - Setor 4 - Prédio 43424 - Sala 116
Bairro Agronomia - CEP 91.509-900 - Porto Alegre - RS
Fone/fax: (51) 3316-6835/(51) 3316-7142
http://www.sbc.org.br

CIP - Biblioteca Setorial de Informática da PUC-Rio

Simpósio Brasileiro de Sistemas Colaborativos (3.: 2006, Natal, RN)
S612 Anais do III Simpósio Brasileiro de Sistemas Colaborativos, Natal, RN, 20 a 22 de novembro de 2006 / editores Alberto Barbosa Raposo e Flávia Maria Santoro. - Porto Alegre: Sociedade Brasileira de Computação, 2006.
ix, 171 p.

SBSC 2006
Realização: Universidade Federal do Rio de Janeiro, Pontifícia Universidade Católica do Rio de Janeiro, Faculdade Filosofia, Ciências e Letras de Ribeirão Preto / USP, Universidade Federal do Rio Grande do Norte.
ISBN 85-7669-097-7

1. Cooperação – Processamento de Dados - Congressos. 2. Grupos de Trabalho– Processamento de Dados– Congressos. I. Raposo, Alberto Barbosa. II. Santoro, Flávia Maria. III. Universidade Federal do Rio de Janeiro. IV. Pontifícia Universidade Católica do Rio de Janeiro. V. Universidade de São Paulo. Faculdade de Filosofia, Ciências e Letras. VI. Universidade Federal do Rio Grande do Norte.

CDD: 004.3306

Apresentação

CSCW (*Computer-Supported Cooperative Work*) é uma área de pesquisas cujo foco de estudos está no uso de computadores como suporte à colaboração entre pessoas. Estes estudos abrangem teoria e prática do trabalho colaborativo, projeto, desenvolvimento e implantação de sistemas colaborativos e relação entre aspectos sociais e técnicos. Por lidar com tecnologia e interação entre humanos, a área é necessariamente multidisciplinar, envolvendo profissionais das áreas de computação, automação, antropologia, sociologia, psicologia social, economia, teoria organizacional, educação, e de outras áreas interessadas no estudo do trabalho colaborativo.

O III Simpósio Brasileiro em Sistemas Colaborativos (SBSC 2006) teve suas origens na organização da Trilha de CSCW no contexto do WebMedia 2003, em Salvador (BA). Dado o número e natureza das submissões encaminhadas à trilha, foi possível identificar uma comunidade de pesquisa em CSCW que precisa ser atendida em suas necessidades de discussões e troca de experiências em âmbito nacional. A partir desta constatação, foram organizados, nos dois anos subseqüentes, o Workshop Brasileiro de Tecnologias para Colaboração (WCSCW). O I WCSCW 2004 foi novamente realizado em conjunto ao WebMedia, em Ribeirão Preto (SP) e o II WCSCW 2005 foi realizado junto ao Simpósio Brasileiro de Informática e Educação (SBIE), em Juiz de Fora (MG).

Em reunião durante o WCSCW 2005, a comunidade presente criou a Comissão Especial em Sistemas Colaborativos (CESC) junto à SBC. A CESC sugeriu a ampliação do evento, que passou a ser denominado Simpósio Brasileiro em Sistemas Colaborativos (SBSC), mantendo o rigor científico, a multidisciplinaridade, a abertura e a transparência, características da CESC.

Em 2006, temos a honra de coordenar o comitê de programa do III SBSC, o primeiro com o novo nome e regido pela CESC, que se firma como o principal fórum no país na área de sistemas colaborativos. O número de submissões vem crescendo continuamente, não só quantitativamente, mas também qualitativamente. Este ano houve 45 submissões, cada uma avaliada por 3 revisores. Os 17 artigos melhor avaliados foram selecionados para publicação.

Este ano o SBSC ocorre em conjunto com o IHC e o WebMedia em Natal (RN). Agradecemos ao trabalho dos Profs. Jair Leite e Raquel Prates, que tornaram isso possível. A realização conjunta dos eventos permitiu, pela primeira vez, a realização de minicursos. Foram 38 submissões, dos quais 6 foram selecionados.

Agradecemos a todos os autores que submeteram seus artigos e minicursos, aos membros do comitê de programa e aos revisores, por terem dedicado seu tempo ao SBSC 2006. Nossos especiais agradecimentos aos Profs. Renata Araujo e Cléver Farias, coordenadores da organização do evento e aos Profs. Hugo Fuks e Marcos Borges, coordenadores da CESC no período 2005-2006.

Agradecemos também aos palestrantes convidados: Profs. Roberto Montezuma (UFPE), Bonnie Nardi (Universidade da California, Irvine, EUA), Alan Dix (Universidade de Lancaster, Reino Unido) e Darko Kirovski (Microsoft Research, EUA).

Desejamos que o SBSC 2006 seja um proveitoso fórum de debates e reflexão sobre os temas mais importantes na área de sistemas colaborativos.

Natal, novembro de 2006.

Alberto Raposo e Flávia Santoro
Coordenadores do Comitê de Programa

SBSC 2006 – III Simpósio Brasileiro de Sistemas Colaborativos

Coordenação Geral:

Cléver Farias (FFCLRP/USP)
Renata Araujo (UNIRIO)

Coordenação do Comitê de Programa:

Alberto Raposo (PUC-Rio)
Flavia Santoro (UNIRIO)

Comitê de Programa:

Alberto Raposo (PUC-Rio)
Ana Carolina Salgado (UFPE)
André Luís Vasconcelos Coelho (UNIFOR)
Autran Macêdo (UFU)
Cláudia Lage Rebello da Motta (UFRJ)
Cleidson de Souza (UFPA)
Cléver Farias (FFCLRP/USP)
Flavia Santoro (UNIRIO)
Hugo Fuks (PUC-Rio)
Jacques Wainer (UNICAMP)
Jano de Souza (UFRJ)
Jauvane Cavalcante de Oliveira (LNCC)
José Maria N. David (FRB)
José Valdeni de Lima (UFRGS)
Karin Becker (PUC-RS)
Marco Aurélio Gerosa (PUC-Rio)
Marco A. S. Mangan (PUCRS)
Marcos R.S. Borges (UFRJ)
Mariano Pimentel (PUC-Rio)
Neide dos Santos (UERJ)
Renata Araujo (UNIRIO)
Rogério Depaula (INTEL)

Avaliadores

Alberto Raposo
Amaury André
Ana Carolina Salgado
André Coelho
Antônio Barros Neto
Autran Macêdo
Borje Karlsson
Cesar T. Pozzer
Cláudia Motta
Cleidson de Souza
Clever Farias
Elmário Dutra Jr.
Enio Russo
Felipe Carvalho
Flávia Santoro
Hugo Fuks
Ivan Ricarte
Jacques Wainer
Jano Souza
Jauvane Oliveira
José Maria David
José Valdeni de Lima
Kelly Hannel
Luciana Lima
Luiz Gonzaga da Silveira Jr
Marco Mangan
Marco Aurélio Gerosa
Marcos Borges
Mariano Pimentel
Neide Santos
Patrícia Tedesco
Pedro Porfirio Farias
Renata Araujo
Roberto Siva Filho
Rodrigo Rech
Rodrigo Reis
Rogério Depaula
Sandra Siebra
Vaninha Vieira

SBC – Sociedade Brasileira de Computação

Presidência:

Presidente: Cláudia Maria Bauzer Medeiros (UNICAMP)
Vice-presidente: José Carlos Maldonado (ICMC - USP)

Diretoria:

Administrativa e Finanças: Carla Maria Dal Sasso Freitas (UFRGS)
Eventos e Comissões Especiais: Karin Breitmann (PUC-Rio)
Educação: Edson Norberto Cáceres (UFMS)
Publicações: Marta Lima de Queiros Mattoso (UFRJ)
Planejamento e Programas Especiais: Virgílio Augusto Fernandes Almeida (UFMG)
Secretarias Regionais: Aline dos Santos Andrade (UFBA)
Divulgação e Marketing: Altigran Soares da Silva (UFAM)
Regulamentação da Profissão: Roberto da Silva Bigonha (UFMG)
Eventos Especiais: Carlos Eduardo Ferreira (USP)

Conselho:

Mandato 2003-2007:
Flávio Rech Wagner (UFRGS)
Siang Wu Song (USP)
Luiz Fernando Gomes Soares (PUC-Rio)
Ariadne Maria B. Carvalho (Unicamp)
Taisy Silva Weber (UFRGS)

Mandato 2005 – 2009:
Ana Carolina Salgado (UFPE)
Ricardo de Oliveira Anido (UNICAMP)
Jaime Simão Sichman (USP)
Daniel Schwabe (PUC-Rio)
Marcelo Walter (UFPE)

Suplentes - Mandato 2005-2007:
Robert Carlisle Burnett (PUC/PR)
Ricardo Reis (UFRGS)
José Valdeni de Lima (UFRGS)
Raul Sidnei Wazlawick (UFSC)

CESC – Comissão Especial de Sistemas Colaborativos

Comitê Gestor:

Coordenador: Hugo Fuks (PUC-Rio)
Coordenador Adjunto: Marcos Borges (UFRJ)

Membros 2005-2006:
Alberto Raposo (PUC-Rio)
Cláudia Motta (UFRJ)
Cléver Farias (FFCLRP/USP)
Flávia Santoro (UNIRIO)
Jacques Wainer (UNICAMP)

Sumário

Um Estudo Etnográfico do Processo de Previsao de Tempo
Weverton Cordeiro, Rafael Esteves, Breno França, Cleidson de Souza 1

Método Colaborativo de Observação: Entendendo coletivamente os ambientes complexos
Renata Machado, Marcos Borges, José Orlando Gomes 10

A Process for User Interaction Analysis in Collaborative Environments
Sandra Siebra, Ana Carolina Salgado, Patrícia Tedesco 19

Avaliando os Elementos de Contexto para as Atividades de Coordenação em Pré-reuniões
Márcio Rosa, José Maria David, Flávia Santoro, Marcos Borges 29

Um Sistema de Recomendação de Especialistas Sensível ao Contexto para Apoio à Colaboração Informal
Helo Petry, Vaninha Vieira, Patrícia Tedesco, Ana Carolina Salgado 38

Percepção de Contextos de Atividade para Identificação de Comunidades Potenciais
Eliane De Bortoli, Cesar Augusto Tacla,, Fabrício Enembreck 48

Modelo 3C de Colaboração para o desenvolvimento de Sistemas Colaborativos
Mariano Pimentel, Marco Aurélio Gerosa, Denise Filippo, Alberto Raposo, Hugo Fuks, Carlos Lucena 58

Um Ambiente para Integração de Aplicações Colaborativas
Roberta Gomes, Guillermo Hoyos-Rivera, Jean-Pierre Courtiat 68

Gestão de Conhecimento sobre Avaliações de Sistemas Colaborativos
Renata Araujo, Flávia Santoro, Fernando Prata, Thalita Moraes 78

PatternFlow: Supporting Standardized Description and Enactment of Business Processes
Marcos Kalinowski, Marcos Borges, Guilherme Travassos 87

Selecting Anomalous Traces from Workflow Event Logs
Fábio Bezerra, Jacques Wainer 98

Distribuição de Tarefas por meio da Seleção Induzida de Recursos
Rogério Silva, Autran Macêdo, Ilmério Reis da Silva, João Souza 107

ThreadMap: Uma Abordagem para Mapeamento de Mensagens de Correio Eletrônico
Hélio Queiroz Jr, Manoel Mendonça 118

NegoSys: um ambiente de suporte à negociação
Melise Paula, Jano Souza 129

Alterando o Grau de Colaboração de uma Ferramenta de Compartilhamento de Desktop por Meio de Plugins
Eduardo Carlos, Alberto Raposo 140

Desenvolvimento de Aplicações de Democracia Eletrônica em Larga Escala Utilizando Grades Computacionais
Rogério Rondini, Hermes Senger, Cléver Farias 150

Simbiose Digital e Colaboração em Ambientes 3D
Glauco Todesco, Marcelo Zuffo 161

Um Estudo Etnográfico do Processo de Previsão de Tempo

Weverton Cordeiro, Rafael Esteves, Breno França, Cleidson R. B. de Souza
Departamento de Informática
Universidade Federal do Pará
Campus Universitário do Guamá, Rua Augusto Côrrea, 01
66075-110 – Belém - Pará - Brasil
wevertoncordeiro@acm.org, {esteves, bfranca, cdesouza} @ufpa.br

ABSTRACT
This paper presents the results of a field study applied to weather forecasting using an ethnographic approach. Our goal was to achieve an understanding of the collaborative aspects that employees use to get their work done. Through the analysis of the collected data, we suggest requirements for a software system to be developed to help these actors work together.

RESUMO
Este artigo apresenta os resultados de um estudo de campo aplicado ao processo de previsão do tempo utilizando uma abordagem etnográfica. O principal objetivo era entender os aspectos colaborativos que os empregados utilizam para cumprir com seus trabalhos. Através da análise dos dados coletados, foram sugeridos requerimentos para um sistema de software a ser desenvolvido para auxiliar esses empregados nas suas tarefas colaborativas.

Termos Gerais
Human Factors.

Palavras-Chave
Etnografia, Engenharia de Requisitos, Meteorologia, Previsão do Tempo.

Keywords
Ethnography, Requirements Elicitation, Weather Forecast.

1. INTRODUÇÃO
Um aspecto importante da área de Trabalho Cooperativo Apoiado por Computador é o entendimento da forma como as pessoas colaboram para a realização de seu trabalho. Isto inclui a interação entre os membros de um grupo para alcançar um objetivo comum, as práticas necessárias para que o trabalho seja executado, as convenções formais e informais adotadas durante o trabalho, entre outros aspectos. Tal entendimento é necessário para o projeto de sistemas computacionais que facilitem o trabalho cooperativo. Desta forma, é imperativa a importância de se considerar aspectos sociais no projeto de softwares colaborativos, conforme inicialmente proposto por Suchman [1]. De fato, a adoção da etnografia no trabalho pioneiro de Suchman influenciou enormemente a área e deu início a uma série de estudos etnográficos visando entender o trabalho realizado por diferentes profissionais. Por exemplo, Heath e Luff [2] identificaram importantes mecanismos de coordenação utilizados por controladores do metrô de Londres. De maneira similar, Bentley e colegas [3] estudaram controladores de trafego aéreo. Grinter [4], por outro lado, discute como engenheiros de software utilizam ferramentas de gerência de configuração para coordenar seu trabalho. Finalmente, Harper [5] discute como o papel não pode ser facilmente substituído por ferramentas computacionais devido às suas características físicas que favorecem a colaboração entre economistas do Fundo Monetário Internacional.

Estes estudos e diversos outros demonstram que a etnografia é uma importante ferramenta na área de CSCW, pois permite a identificação de requisitos que dificilmente seriam obtidos apenas observando as regras que devem ser aplicadas em uma organização. Estes requisitos existem pois em qualquer processo, diferentes nuances e detalhes são necessários para sua adequada realização [6]. A etnografia permite entender como os participantes colaboram visando atingir um objetivo comum e desta forma revela aspectos existentes em ambientes cooperativos que *devem* ser levados em consideração no projeto de um sistema computacional. Em resumo, a etnografia, no contexto de CSCW, pode ser vista como uma ferramenta que permite o desenvolvimento de sistemas computacionais adequados ao contexto sócio-técnico onde estão inseridos.

O papel da etnografia na área de CSCW tem sido amplamente discutido na literatura. De um modo geral, entre os resultados que um estudo etnográfico pode gerar, cabe destacar:

- A identificação de fluxos de informação não documentados (isto é, a ordem na qual a informação é transmitida, a mídia utilizada para a transmissão da informação, os protocolos que governam esta transmissão, etc.);
- A identificação da real divisão do trabalho, uma vez que a divisão é organizada de forma dinâmica, geralmente contrariando ou flexibilizando os organogramas estabelecidos, e que, portanto, não necessariamente seguem uma forma prescrita;
- Finalmente, o responsável pela condução do estudo etnográfico pode representar o usuário final do sistema a ser desenvolvido, devido ao vasto conhecimento adquirido a respeito do processo durante o período de realização do estudo etnográfico. Isso permite que o projeto e

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 1-9.

desenvolvimento do sistema possam ser realizados de maneira mais simples e menos intrusiva: perguntas simples podem ser respondidas pelo etnógrafo sem interromper o usuário.

O presente artigo apresenta os resultados de um estudo etnográfico que teve como objetivo entender a forma como o trabalho é realizado pelos meteorologistas do 2º Distrito de Meteorologia em Belém-PA (2º DISME). Neste trabalho procurou-se identificar aspectos implícitos relacionados às práticas do dia-a-dia dos indivíduos envolvidos no processo principalmente no que se refere à colaboração. O objetivo final deste estudo é a identificação de requisitos para uma ferramenta de apoio ao trabalho de meteorologistas.

O restante do artigo está organizado como segue. A metodologia adotada neste trabalho é descrita na seção 2. Nas seções 3 e 4, o ambiente e o processo de previsão do tempo são descritos. Sugestões de requisitos de softwares que consideram os aspectos específicos do trabalho dos meteorologistas são apresentadas na Seção 5. Finalmente, a Seção 6 conclui o artigo e apresenta sugestões para trabalhos futuros.

2. METODOLOGIA

O estudo etnográfico descrito neste artigo corresponde a um estudo de campo que visa maximizar o realismo do contexto no qual as evidências foram adquiridas [7]. Deste modo, para a coleta de dados foram realizadas diversas visitas ao local de estudo para observar as práticas e o fluxo de trabalho das pessoas envolvidas no processo de previsão do tempo. As visitas iniciais foram realizadas com o objetivo de adquirir um entendimento geral sobre o processo de geração de previsões do tempo, desde a coleta de dados até à previsão propriamente dita. Para isso, foram conduzidas entrevistas semi-estruturadas e não estruturadas [8] com os envolvidos no processo. Em seguida, partiu-se para a etapa de observações [9], onde o fluxo de eventos foi registrado em notas de campo. Os autores se alternaram nas observações nos diversos setores que participam do processo de previsão. As observações ocorreram durante um período de um mês, em um total de 12 visitas ao local de trabalho dos meteorologistas do 2º DISME e quatro visitas à duas estações climatológicas controladas pelo 2º DISME. As visitas corresponderam a um total de 52 horas de pesquisa de campo. Um total de dez entrevistas foram realizadas distribuídas da seguinte forma: três observadores alocados nas estações climatológicas de superfície das cidades de Belém-PA e Itaituba-PA, dois operadores de telecomunicação (NUTEL), quatro meteorologistas (SEPRE), e o meteorologista coordenador do 2º DISME. Cada uma das entrevistas durou entre 45 e 60 minutos.

Análises e interpretações dos dados coletados (notas de campo e entrevistas transcritas) para identificar aspectos importantes do trabalho de previsão de tempo foram realizadas. Neste caso, o principal objetivo era identificar mecanismos que poderiam ser adotados de modo a facilitar o trabalho dos meteorologistas do 2º DISME. Além disso, a análise também visou a identificação de requisitos para um ambiente computacional de auxílio ao trabalho dos meteorologistas. Uma vez que estes requisitos foram identificados, novas entrevistas com os meteorologistas foram conduzidas visando validá-los. Os meteorologistas do 2º DISME confirmaram e/ou modificaram as sugestões propostas. Os resultados descritos neste artigo são aqueles já validados pelos meteorologistas.

3. O AMBIENTE ESTUDADO

O 2° DISME é distrito do INMET (Instituto Nacional de Meteorologia), responsável por coletar os dados meteorológicos provenientes de estações climatológicas situadas em localidades pertencentes aos estados do Pará, Amapá e Maranhão e, com esses dados, gerar previsões de tempo para essa região. Para os objetivos deste estudo, este órgão é formado essencialmente pelo NUTEL (Núcleo de Telecomunicações) e o SEPRE (Serviço de Previsão). O DISME também possui outros setores responsáveis por questões administrativas (controle de funcionários, controle de material, etc.) e que, portanto, não foram considerados neste estudo.

O 2º DISME controla um total de 29 estações climatológicas de superfície, sendo 27 convencionais – que necessitam de um operador – e 2 automáticas – nas quais a coleta e transmissão dos dados meteorológicos ocorrem sem intervenção humana. Os dados das estações climatológicas convencionais são enviados para o NUTEL, que recebe os dados das observações e os repassa para o SEPRE, responsável por elaborar a previsão do tempo propriamente dita.

Trabalham no 2º DISME um total de 96 pessoas, sendo 70 servidores, 5 estagiários e 21 funcionários terceirizados. Um total de 47 pessoas trabalham nas estações climatológicas de superfície, com uma média de 2 observadores por estação, que trabalham em turnos. O NUTEL possui 7 funcionários que são escalados de acordo com os três horários do dia destinados à recepção dos dados. Somente um operador recebe os dados, sendo que o mesmo operador pode trabalhar em mais de uma recepção no mesmo dia. O Serviço de Previsão conta com 5 meteorologistas e 1 auxiliar sendo que apenas um meteorologista é responsável pela elaboração previsão do tempo de cada dia.

O NUTEL e o SEPRE estão localizados em salas distintas no 2º DISME, conforme ilustrado na Figura 1. Geralmente só uma pessoa trabalha no NUTEL. No SEPRE existe uma variação no número de pessoas que habitam a sala. Normalmente encontra-se o meteorologista responsável pela previsão daquele dia, um auxiliar, e outro meteorologista responsável por outras tarefas não relacionadas à previsão em si.

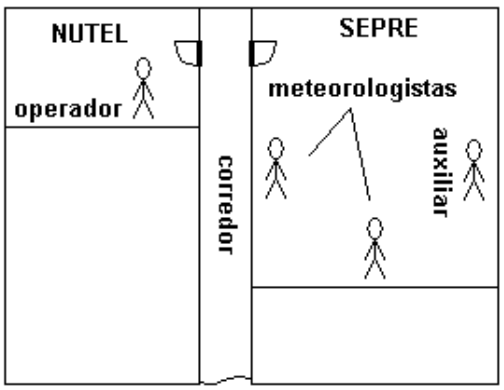


Figura 1. Organização dos Setores de Previsão do Tempo do 2º DISME

4. O PROCESSO DE PREVISÃO DO TEMPO

O estudo etnográfico realizado permitiu a identificação dos diferentes componentes organizacionais (setores, núcleos, etc.)

envolvidos no processo de elaboração da previsão do tempo. Após esta identificação, o foco deste estudo visou o entendimento do funcionamento interno de cada um destes componentes. A partir de uma série de observações, foi elaborado o diagrama que reflete o fluxo de informações que ocorre no 2º DISME, apresentado na Figura 2.

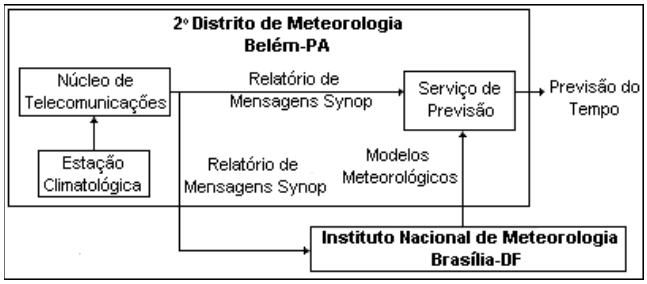


Figura 2. Fluxo de Informações do 2º DISME

O processo de previsão acontece da seguinte forma: em três horários distintos do dia (12:00 GMT, 18:00 GMT e 00:00 GMT, respectivamente 09:00 hs, 15:00 hs e 21:00 hs, horário de Brasília) os dados são transmitidos das estações climatológicas através de telefone – no caso das estações climatológicas de superfície convencionais – para o NUTEL (órgão centralizador das informações em Belém-PA). Este órgão envia as informações para Brasília-DF para a elaboração da previsão do tempo em escala continental. Os dados também são analisados por meteorologistas do Serviço de Previsão (SEPRE) para a elaboração da previsão regional. As estações climatológicas de superfície automáticas coletam e enviam seus dados constantemente para Brasília e atualmente não são considerados durante o processo de elaboração da previsão do tempo pelo 2° DISME.

As próximas subseções têm o objetivo de apresentar com uma maior riqueza de detalhes cada etapa do processo de previsão do tempo. Mais especificamente serão apresentados os esquemas de funcionamento interno das estações climatológicas localizadas em Belém e Itaituba (PA), do Núcleo de Telecomunicações (NUTEL) e do Serviço de Previsão (SEPRE), os quais são apresentados na Figura 2, bem como o fluxo de informações entre eles.

4.1. A Coleta de Informações Meteorológicas

O processo de elaboração da previsão meteorológica referente ao período de 24 horas do próximo dia tem início com a coleta dos dados em cada uma das estações climatológicas de superfície espalhadas pelo Brasil.

As observações meteorológicas na estação climatológica de superfície são realizadas diariamente, em três períodos diferentes: às 12:00 GMT, às 18:00 GMT e às 00:00 GMT. Essas observações são divididas, em um único dia, por *até* dois observadores diferentes que trabalharão em horários diferentes. Nos dias em que as observações são divididas entre dois observadores, o observador que efetuará a observação das 12:00 GMT também realizará a observação das 00:00 GMT. Essa divisão dos horários não é uma regra, mas permite que o meteorologista que fez a primeira leitura de alguns aparelhos possa ter uma visão mais nítida da tendência observada durante o dia, quando o mesmo realizar a última leitura do dia.

O trabalho do observador na estação climatológica inicia com a coleta dos dados meteorológicos nos aparelhos. Entre os dados colhidos estão à pressão atmosférica, umidade relativa do ar, temperatura do ar, quantidade de chuva, temperatura do ponto de orvalho, etc. Em seguida, os dados são analisados, transferidos para uma carta e em seguida transmitidos para o NUTEL, via telefone. O mesmo fluxo de trabalho ocorre em cada uma das 27 estações convencionais.

Existem algumas regras que devem ser seguidas pelo observador durante a coleta dos dados meteorológicos, com o objetivo de não provocar perturbações nos dados coletados. Por exemplo, os termômetros de máxima e de mínima devem ser lidos 10 minutos antes do envio da mensagem para o NUTEL, sendo que a transmissão da mensagem deve ser realizada exatamente na *hora cheia* (termo utilizado pelos meteorologistas para indicar a hora em ponto). Outros aparelhos observados incluem os termômetros de bulbo seco e úmido, o hidrógrafo, o termógrafo, o termidrografo, o pluviografo, o tanque de evaporação, o heliógrafo e o anenografo.

Outro procedimento executado pelo observador é a identificação de nebulosidade. Um dos meteorologistas informou que para descrever a nebulosidade ele divide mentalmente o céu em dez partes e aplica uma série de heurísticas para determinar a nebulosidade. Por exemplo, se sete das dez partes em que o céu foi dividido apresentarem o mesmo tipo de nuvem, então apenas será considerado aquele tipo, em detrimento dos outros tipos que também podem ser observados. Um outro hábito específico refere-se à verificação da visibilidade no momento da coleta dos dados: um outro meteorologista informou que o cálculo da visibilidade é feito observando-se a parte do horizonte que possui a maior densidade de nuvens, e a partir da observação fixa para esta parte do horizonte, estima a distância em metros até onde existe uma boa visibilidade.

Tendo terminado a coleta dos dados, o observador se dirige até o prédio da estação e efetua a transformação de alguns dados, utilizando mapas, manuais, escalas e outras anotações em papel. Um exemplo de uma transformação de dados realizada pelo meteorologista da estação climatológica diz respeito a representação da pressão atmosférica, a qual deve ser expressa em relação ao nível do mar. Outra transformação realizada é em relação à velocidade do vento: lida do anemômetro em quilômetros por hora, mas expressa em nós (unidade de medida de velocidade equivalente a uma milha marítima por hora) na carta.

Todas as transformações de dados realizadas têm por objetivo a elaboração da mensagem Synop. A mensagem Synop, *Surface Synoptic Observation* [10], é um código numérico utilizado para reportar observações meteorológicas feitas por estações manuais ou automáticas. Seu objetivo principal é facilitar a transmissão da mensagem através de ondas curtas de rádio e o entendimento da mensagem pelas pessoas que entendam o código. Como a mensagem Synop constitui um formato internacional de comunicação de dados meteorológicos, as unidades de medidas utilizadas para representar a velocidade do vento, pressão atmosférica, etc., devem ser as unidades de medidas padrão do Sistema Internacional, tornando assim necessária a transformação dos dados mencionada anteriormente.

Uma mensagem Synop consiste de grupos de números (ou barras quando a informação não está disponível) descrevendo informações gerais sobre as condições atmosféricas, tais como temperatura, pressão barométrica e visibilidade na estação onde foi realizada a leitura. A Tabela 1 a seguir apresenta a mensagem

enviada da estação 82445 (localizada em Itaituba, Estado do Pará) para o NUTEL em 08 de março de 2006, às 12:00 GMT.

Tabela 1. Exemplo de mensagem Synop

```
08124 82445 21596 80000 10244 20239
30094 40147 70365 8452/ 333 20234 59004
60074 555 60060
```

A mensagem Synop apresentada na Tabela 1 permite identificar algumas peculiaridades da mensagem. O primeiro grupo de cinco algarismos informa o dia em que a mensagem foi elaborada (08), a hora a qual se refere a mensagem (12, de 12:00 GMT) e o algarismo 4 é um algarismo fixo, para controle. O grupo de cinco algarismos subseqüente é o identificador da estação climatológica a qual se refere à mensagem. Os dois primeiros algarismos deste grupo identificam o distrito ao qual a estação climatológica pertence (2º DISME - 82). Os números que aparecem destacados nos grupos subseqüentes ao identificador da estação climatológica, na mensagem Synop, indicam um identificador de grupo. Os grupos **333** e **555** são utilizados para denotar, respectivamente, pausa e fim da mensagem iminente. O caractere '/' presente na mensagem indica que a informação não está disponível devido à ausência do aparelho necessário para fornecer a informação.

Os dados coletados de todas as estações climatológicas de superfície do país, bem como de outras estações climatológicas de superfície espalhadas ao redor do mundo, permitirão montar a carta de superfície, um instrumento importante para o processo de elaboração da previsão do tempo, conforme discutido posteriormente no artigo.

A última etapa do processo de coleta de informações meteorológicas consiste em repassar a mensagem Synop para o NUTEL do DISME ao qual a estação está subordinada. Este processo será descrito na próxima seção.

4.2. A Recepção dos Dados das Estações Climatológicas

Cada DISME possui um NUTEL, o qual é responsável por receber as mensagens Synop de todas as estações climatológicas daquele distrito, submeter estas informações ao sistema computacional *INMET – Sistema Nacional de Informação Hidro-Meteorológica*, e repassar um relatório contendo todas as mensagens Synop coletadas para o SEPRE. O SEPRE é o setor efetivamente responsável pela elaboração da previsão do tempo.

O período de serviço no NUTEL acompanha o período de serviço dos observadores nas estações climatológicas. Portanto, são definidas três escalas de serviços no NUTEL: às 12:00 GMT, às 18:00 GMT e às 00:00 GMT. Ao final de cada escala, que tem duração de 30 minutos, o operador realiza duas tarefas: enviar os dados recebidos das estações climatológicas para o INMET e imprimir um relatório contendo todas as mensagens recebidas para repassar ao SEPRE.

Em cada uma das escalas, apenas uma pessoa trabalha recepcionando os dados. O responsável na escala atual possui a sua disposição um computador, dois telefones (um em cada lado), papel, caneta e impressora, conforme ilustrado pela Figura 3.

Figura 3. Operador do NUTEL Atendendo o Observador de uma Estação Climatológica.

Durante um período de 30 minutos, cada estação climatológica liga para um dos telefones do NUTEL e transmite a mensagem Synop a ser registrada no sistema. Ao todo, 27 estações transmitem suas mensagens no período de meia hora. Cada transmissão dura em média 45 segundos. Quando uma estação liga para o NUTEL e o operador está ocupado com uma outra transmissão, o operador atende ao telefone, porém o deixa de lado, até que a transmissão da mensagem corrente seja finalizada, conforme pode ser observado na Figura 3.

O processo de transmissão da mensagem Synop utiliza o código Q de comunicação [11], cuja função é simplificar, dar maior fluidez e principalmente proporcionar um maior entendimento entre operadores de radiocomunicação em qualquer idioma, sejam utilizando a linguagem falada ou utilizando linguagem codificada em Código Morse. No código Q, as frases que ocorrem freqüentemente na comunicação entre duas pessoas são substituídas por um conjunto de três letras, sempre iniciadas pela letra Q. Alguns códigos utilizados na comunicação entre o operador do NUTEL e o observador da estação climatológica podem ser observados na Tabela 2.

Tabela 2. Alguns códigos de Comunicação Utilizados.

Código	Mensagem	Código	Mensagem
QAP	Estou na Escuta	QRX	Aguarde um Pouco
QRA	Nome	QSL	Tudo Entendido
QRM	Ruído na Comunicação	QSM	Repita a Mensagem (RPT)
QRS	Transmitir mais Devagar	QTA	Cancele Mensagem Anterior

A primeira etapa de transmissão da mensagem Synop é a identificação da cidade onde a estação se localiza. Em seguida, o observador informa o código identificador da estação, o qual é composto por cinco algarismos. Depois disso, o observador informa ao operador os dados da mensagem Synop. A forma como a mensagem é transmitida também possui um caráter peculiar: os números correspondentes ao código da estação são pronunciados como algarismos indo-arábicos; os números

correspondentes ao conteúdo da mensagem são transmitidos como algarismos ordinais, exceto os números de controle, os quais são transmitidos como algarismos indo-arábicos. A Tabela 3 ilustra a conversa ocorrida entre o operador do Nutel e o meteorologista da cidade de Itaituba-PA, durante a transmissão da mensagem Synop apresentada na Tabela 1.

Tabela 3. Conversa enre o Meteorologista e o Operador do Nutel Durante a Transmissão da Mensagem Synop

Meteorologista (M) - Itaituba.
Operador Nutel (OP) - Pode prosseguir.
(M) - Oito dois quatro quatro cinco
(M) - Segundo primeiro quinto nono sexto
(M) - Oitavo, zero dobrado, zero dobrado
(M) - Primeiro zero segundo quarto dobrado
(M) - Segundo zero segundo terceiro nono
(M) - Terceiro zero dobrado nono quarto
(M) - Quarto zero primeiro quarto sétimo
(M) - Sétimo zero terceiro sexto quinto
(M) - Oitavo quarto quinto segundo diagonal
(M) - Três três três
(M) - Segundo zero segundo terceiro quarto
(M) - Quinto nono zero dobrado quarto
(M) - Sexto zero dobrado sétimo quarto
(M) - Cinco cinco cinco
(M) - Sexto zero dobrado sexto zero
(M) - QSL ?
(OP) - Até mais tarde.
(M) - Até mais.

A medida que a mensagem SYNOP é transmitida, ela é registrada no sistema. O objetivo desse procedimento é enviar, ao fim da escala, todas as mensagens recebidas para o INMET. Após o envio de todas as mensagens ao INMET, um relatório contendo todas as mensagens é impresso e enviado ao SEPRE para a elaboração da previsão do tempo.

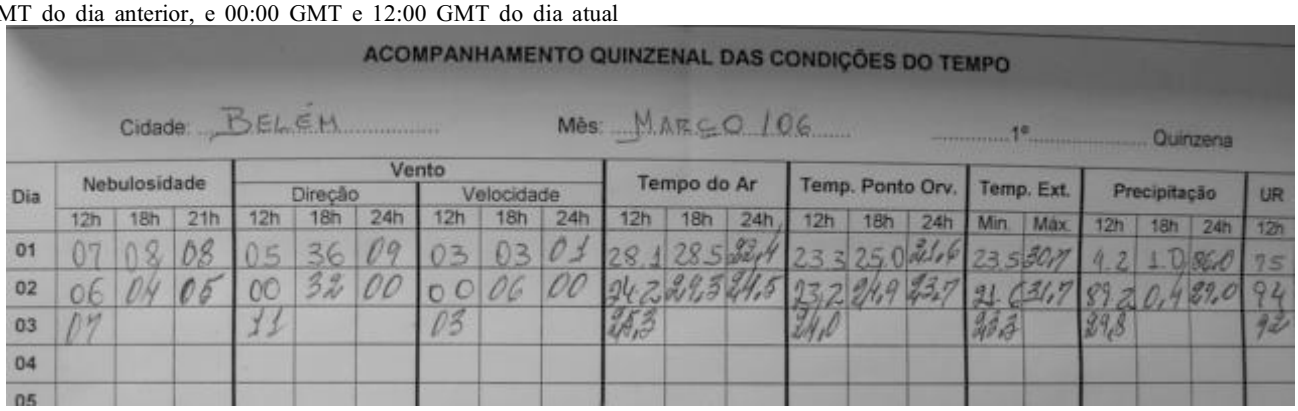
4.3. A Elaboração da Previsão do Tempo

A elaboração da previsão do tempo é a etapa final do processo e consiste na análise e comparação dos dados recebidos para se chegar a conclusões referentes às condições do tempo no dia seguinte. A previsão do tempo utiliza dados coletados às 18:00 GMT do dia anterior, e 00:00 GMT e 12:00 GMT do dia atual para gerar a previsão do dia posterior. Com estes dados, o meteorologista gera a previsão do tempo e a repassa aos órgãos interessados, como a imprensa, por meio do telefone, e-mail, boletins de informação meteorológica disponíveis na Internet.

A partir do momento em que os dados oriundos do NUTEL chegam ao SEPRE, eles são repassados para uma pessoa encarregada de converter as mensagens Synop para um novo formato. Este novo formato visa facilitar a leitura dos dados por parte do meteorologista que irá elaborar a previsão do tempo. Os dados são colocados em planilhas que são divididas de acordo com o tipo de medida apropriada (temperatura, ventos, precipitação, nebulosidade, e assim por diante) e agrupadas de acordo com a região. Vale ressaltar que essa conversão é um procedimento interno no SEPRE do 2º DISME, pois não possui caráter oficial, constituindo-se em um instrumento adotado por estes profissionais para facilitar a realização de seu trabalho. A Figura 4 ilustra um exemplo desse artefato.

Os dados coletados pelas estações meteorológicas que chegam ao SEPRE por meio do NUTEL são conhecidos como dados de superfície (os quais podem ser obtidos na carta de superfície). Além dos dados de superfície, existe um outro tipo de informação que é utilizada na previsão, os dados de altitude (os quais podem ser obtidos na carta de altitude). Dados de altitude são coletados através de balões meteorológicos equipados com sensores e representam medidas de precipitação, trovoadas e outras informações da atmosfera que são relevantes para a previsão.

Outra ferramenta bastante utilizada pelo meteorologista é o modelo meteorológico que pode ser de diversos tipos e conter diferentes tipos de informações. Os modelos meteorológicos permitem a visualização de informações referentes a diversas variáveis como: precipitação, ventos, cobertura de nuvens, temperatura, umidade relativa, etc. Os modelos procuram inferir o comportamento dessas variáveis para as próximas horas e para os próximos dias, desta forma eles são ferramentas de apoio à geração da previsão do tempo. Um dos modelos mais utilizados é o Modelo Brasileiro de Alta Resolução (MBAR) [8]. A Figura 5 ilustra o MBAR representando a variável precipitação.



ACOMPANHAMENTO QUINZENAL DAS CONDIÇÕES DO TEMPO

Cidade: BELÉM Mês: MARÇO 106 1º Quinzena

Dia	Nebulosidade			Vento						Tempo do Ar			Temp. Ponto Orv.			Temp. Ext.		Precipitação			UR
				Direção			Velocidade														
	12h	18h	21h	12h	18h	24h	12h	18h	24h	12h	18h	24h	12h	18h	24h	Min.	Máx.	12h	18h	24h	12h
01	07	08	08	05	36	09	03	03	01	28.1	28.5	22.4	23.3	25.0	21.6	23.5	30.7	4.2	1.0	86.0	75
02	06	04	05	00	32	00	00	06	00	24.2	29.5	24.5	23.2	24.9	23.7	21.6	31.7	89.2	0.4	89.0	94
03	07			11			03			25.3			24.0			22.7		29.8			92
04																					
05																					

Figura 4. Dados Convertidos para Auxílio na Previsão do Tempo.

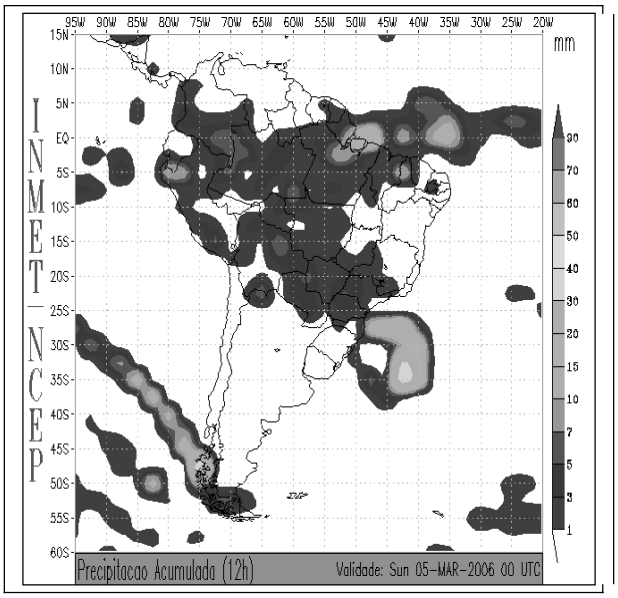


Figura 5. MBAR Mostrando a Precipitação no Continente Sul-Americano.

Imagens de satélite também são utilizadas pelos meteorologistas em seu trabalho. Essas imagens apresentam dados gerados em intervalos de tempo regulares e representam variáveis como nível de vapor de água, temperatura das nuvens, calor, etc. A Figura 6 ilustra uma imagem de satélite que informa o nível de calor emitido para o continente sul-americano.

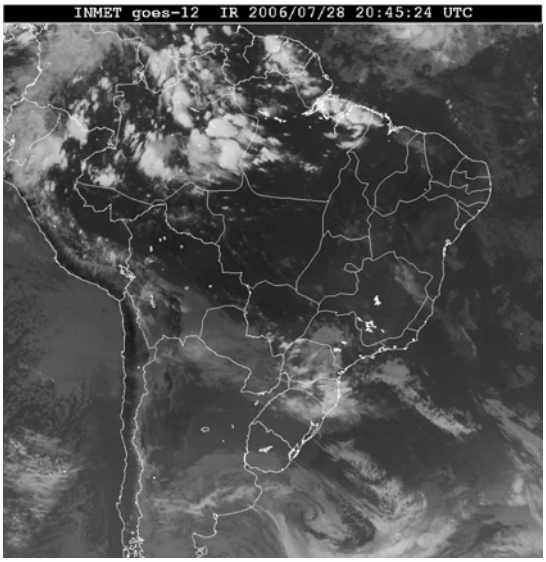


Figura 6. Imagem de Satélite Mostrando a Energia Emitida (Calor) no Continente Sul-Americano.

A principal diferença entre as imagens de satélite e os modelos é que a imagem ilustra fenômenos que já aconteceram ou que estão acontecendo no momento, enquanto que os modelos meteorológicos são prognósticos para curto, médio e longo prazo.

Em resumo, o meteorologista emite seu parecer sobre o comportamento das variáveis meteorológicas no dia seguinte a partir de diversos dados: a partir dos dados coletados pelas estações meteorológicas representadas nas planilhas descritas anteriormente, dos dados contidos nas cartas de altitude e da visualização de diversos modelos meteorológicos e imagens de satélite. Se houver qualquer diferença entre a previsão e as condições reais no dia seguinte, a previsão do tempo deverá ser retificada em um momento posterior.

A experiência do meteorologista influencia de maneira significativa na previsão do tempo que ele elabora. Para ser mais preciso, o conhecimento da região e de suas características em diferentes períodos do ano contribui também para a geração da previsão do tempo. Por exemplo, se o meteorologista sabe que a região sob sua responsabilidade está num período do ano em que são registrados altos índices pluviométricos, ele considerará alta a probabilidade de que chova para essa região e apenas considerará a redução da probabilidade caso os dados e modelos meteorológicos mostrem uma *forte* tendência de mudança.

No SEPRE existem quatro computadores à disposição do meteorologista e cada um pode estar exibindo ou não um modelo meteorológico conforme ilustrado na Figura 7. As previsões são cadastradas em um sistema que as enviará para um ponto central em Brasília-DF. Finalmente, o meteorologista gera previsões simplificadas para outros dias além do dia seguinte – denominadas prognósticos – e os envia para órgãos interessados, por exemplo a imprensa local.

Figura 7. Modelos a Disposição do Meteorologista da Escala.

Prognósticos podem ser de curto prazo (6 a 24 horas), médio (72 a 96 horas) e longo prazo (30 a 90 dias) e visam atender necessidades específicas de organizações e da sociedade em geral. Por exemplo, uma pessoa pode querer saber como estará o tempo a curto prazo, para decidir se irá à praia ou não, ou mesmo para decidir se um determinado evento será realizado ao ar livre ou não; uma indústria de refrigerantes pode solicitar prognósticos de temperatura para determinar se a produção de refrigerantes deve ser aumentada ou reduzida para os próximos dias; também é interessante para a companhia de energia elétrica saber onde poderão ocorrer chuvas intensas, para que assim possa deixar suas equipes de prontidão e acioná-las rapidamente em caso de algum incidente ocasionado por chuvas ou raios. Os prognósticos também são importantes para a agricultura, pois permitem ao produtor tomar decisões em relações as culturas a serem desenvolvidas de acordo com o volume de água esperado para a região para os próximos meses.

A seguir, um exemplo de previsão do tempo é apresentado para a cidade de Belém no dia 31 de julho de 2006.

Segunda, 31 de julho de 2006

CLARO A PARCIALMENTE NUBLADO PASSANDO A NUBLADO.

TEMPERATURA: ESTÁVEL MAX.: 33°C MIN.: 23°C

VENTO DIREÇÃO: SE-E

INTENSIDADE: FRACOS

NASCER DO SOL: 06:18h - OCASO DO SOL: 18:21h

4.4. Observações Importantes

Observações empíricas realizadas pelos próprios meteorologistas do 2º DISME sugerem que, entre outros, o maior índice de acertos na previsão do tempo ocorre quando o mesmo meteorologista trabalha durante vários dias seguidos. O fato de o meteorologista, ao estar envolvido por um maior tempo na previsão do tempo, estar ciente de todas as tendências climatológicas durante aquele período, contribui para o aumento da probabilidade de acertos, uma vez que o meteorologista da escala passa a estar ciente das últimas tendências climáticas.

As práticas utilizadas pelos meteorologistas para minimizar os efeitos da mudança do meteorologista na escala são o chamado "*briefing"* e a observação do trabalho do colega pelo próximo meteorologista da escala. O *briefing* é uma conversa entre o próximo meteorologista da escala e o meteorologista que estava trabalhando anteriormente sobre como o tempo se comportou até o momento da troca de serviço. A observação do trabalho do meteorologista da escala atual pelo próximo meteorologista da escala é realizada de maneira informal, durante as conversas que ocorrem no SEPRE. Esta observação não é uma atividade exclusiva, uma vez que cada meteorologista pertencente ao SEPRE sempre está desempenhando outras atividades no SEPRE. Para observar o trabalho do colega, o meteorologista fica sentado próximo ao auxiliar de meteorologista, conforme disposição apresentada na Figura 1, prestando atenção nos comentários que o meteorologista da escala faz a respeito do seu trabalho, ao mesmo tempo em que conversam informalmente. De maneira análoga, um comportamento típico do meteorologista que está responsável pela previsão do tempo na escala atual é narrar o seu trabalho. Isto é, a medida que a previsão do tempo está sendo elaborada, seus cálculos e observações vão sendo pronunciados em voz baixa. Isso permite que o meteorologista da próxima escala possa acompanhar as tendências climáticas que vão se configurando na região durante o período de trabalho do colega, sem entretanto que ele tenha que se concentrar explicitamente neste acompanhamento. Em alguns momentos o meteorologista da escala chama explicitamente a atenção dos colegas para algum evento atípico observado, como por exemplo, um alto índice pluviométrico em uma determinada cidade.

5. REQUISITOS COMPUTACIONAIS

5.1. Automação de Conversões

O fluxo de dados entre os setores e núcleos que compõem o 2º DISME é em sua maior parte efetuado através de documentos. A importância da troca de dados utilizando esse meio é a de facilitar a verificação de erros ou inconsistências nos dados. Por exemplo, conforme mencionado na seção 4.3, existe uma conversão dos dados Synop vindos do NUTEL para um formato auxiliar que servirá de entrada para o meteorologista produzir a previsão. Esses dados são convertidos em uma planilha e agrupados de acordo com a variável meteorológica e a região. Esta conversão é útil em duas perspectivas diferentes:

- Permite que os dados possam ser passados para os mapas que deverão ser enviados para outros setores, o de controle de qualidade e o de arquivamento de informações;
- Permite que inconsistências nas mensagens Synop possam ser verificadas. Um exemplo típico de inconsistência é a temperatura máxima para o dia estar menor que a temperatura mínima. Um outro exemplo de inconsistência é referente à leitura realizada pelo meteorologista não ser compatível com a série histórica ou com as características climáticas da região – neste caso o meteorologista responsável pela coleta será contatado para maiores esclarecimentos. Por exemplo, a mensagem Synop recebida de uma estação climatológica de superfície informa que houve 12 mm de precipitação (chuva) na cidade. Entretanto o meteorologista do SEPRE que analisou esses dados sabe que na época do ano em que essa leitura foi realizada a precipitação dificilmente alcança valores superiores a 6 mm nessa mesma cidade, ou na região em que essa cidade se encontra. Portanto, o meteorologista responsável pela leitura do pluviografo será contatado para confirmar a informação. Em alguns casos essas inconsistências podem ser indicativas de fraude, isto é, o meteorologista faltou no serviço, e forjou dados para enviar ao NUTEL, de modo que a ausência no trabalho não fosse contabilizada. Como a transmissão dos dados para o NUTEL é feita por telefone, não há nenhuma garantia de que o meteorologista realmente esteve na estação meteorológica para coletar os dados, salvo pela ausência de erros e inconsistências na mensagem Synop.

A partir dessas observações pode-se constatar a importância das conversões manuais realizadas pelos meteorologistas e, portanto, sugere-se cautela no processo automático de conversões de dados meteorológicos. Um software que facilite o trabalho destes profissionais deve permitir a automação de certas conversões (como por exemplo para a geração da mensagem Synop), mas ao mesmo tempo, ele precisa ser flexível para não interromper as convenções existentes no trabalho dos mesmos [12]. A simples possibilidade de tornar uma conversão manual em automática não implica na adoção desta medida, deve-se considerar que esta conversão permite a detecção de erros, portanto, caso ela seja feita de maneira automática, novos mecanismos de detecção devem ser adotados.

5.2. A Experiência dos Meteorologistas

Este mesmo sistema computacional deve fornecer flexibilidade ao meteorologista para que o mesmo possa utilizar a sua experiência a respeito do comportamento climático da região. Segundo o depoimento de um meteorologista:

"*Experiência e conhecimento do comportamento da região são fatores determinantes para a previsão*".

Por exemplo, apesar de o céu estar claro e ensolarado, a previsão do tempo para o resto do dia não necessariamente será sol, pois qualquer meteorologista que trabalhe em Belém, sabe que na época do inverno a probabilidade de ocorrência de chuva pelo período vespertino é extremamente alta.

O conhecimento sobre a região é utilizado pelos meteorologistas na identificação de inconsistências nos dados: um meteorologista entrará em contato com um colega para verificar a precipitação informada de uma cidade em um período somente se ele tiver conhecimento sobre a precipitação média daquela cidade. Se esta experiência for limitada ou inexistente, mecanismos de auxílio devem ser previstos no sistema computacional.

5.3. Coordenação Informal

Durante a realização das atividades do processo de elaboração da previsão do tempo é comum a existência de conversas informais sobre assuntos diversos. Por exemplo, na recepção dos dados das estações, vez ou outra o operador conversa com o observador acerca de outros assuntos não relacionados à transmissão propriamente dita. No entanto, essas conversas podem abordar outros aspectos importantes do trabalho como o estado de tarefas pendentes, solicitação de algum recurso, documento, ou mesmo apenas para contribuir na criação de um senso de comunidade que por sua vez estimula a cooperação. Em outras palavras, as conversas informais podem auxiliar a coordenação das atividades. Desta forma, sistemas colaborativos como "*media spaces*", ambientes colaborativos virtuais e salas de bate papo poderiam ser usados para facilitar a comunicação informal entre meteorologistas.

5.4. "Awareness"

Um sistema computacional para dar suporte ao trabalho dos meteorologistas também precisa fornecer facilidades para que os meteorologistas da escala *atual* tenham ciência do trabalho do meteorologista que trabalhou na escala *anterior*. Para ser mais específico, esta situação ilustra claramente o conceito de percepção (em inglês *awareness*), largamente abordado na área de CSCW [13]: o meteorologista da próxima escala *monitora* o trabalho do meteorologista da escala atual, ao mesmo tempo em que o segundo *informa ou sugere* detalhes relevantes dos dados coletados para o primeiro. Esta situação é basicamente similar àquela identificada no trabalho seminal de Heath e Luff [2] quando estudando controladores de trafego aéreo. Deve-se observar que no caso dos meteorologistas, entretanto, o monitoramento e a apresentação de informações relevantes não precisa ser feita em tempo real, como entre os controladores. Suporte tecnológico para esta característica do trabalho dos meteorologistas poderia ser obtido através do registro e/ou anotação de informações relevantes pelo profissional da escala *de maneira não-intrusiva* para informar ao próximo profissional escalado. O aspecto importante neste caso é o aspecto não-intrusivo da tecnologia para permitir que esta solução seja adotada. Em outras palavras, se o sistema computacional a ser implantado requerer um alto esforço do profissional da escala para indicar aspectos importantes para o colega, este sistema estará fadado ao fracasso [14].

5.5. Suporte Explicito a Colaboração

Conforme mencionado na seção 4, os meteorologistas do SEPRE adquirem os dados a respeito da região sob seu domínio administrativo a partir dos relatórios enviados pelo próprio NUTEL. Em contraste, as informações são necessárias para o trabalho do meteorologista do SEPRE referentes a outras regiões (como por exemplo, informações meteorológicas provenientes das estações climatológicas da Amazônia Ocidental, as quais estão sob o domínio administrativo do 1º DISME, cuja sede é em Manaus-AM), são adquiridas apenas através do sistema central de Brasília-DF. Isto é, não existe nenhum canal permanente de comunicação entre os meteorologistas do SEPRE de Belém e do SEPRE de Manaus, exceto através de e-mails. A troca de informações entre estes distritos é mediada por Brasília. Entretanto, é importante considerar que uma maior troca de informações entre os meteorologistas do SEPRE do 2º DISME com os meteorologistas do SEPRE do 1º DISME seria extremamente benéfica: ambas as regiões, a Amazônia Oriental e a Amazônia Ocidental influenciam significativamente as condições climáticas uma da outra. Atualmente, existe apenas uma videoconferência com hora marcada entre todos os meteorologistas do Brasil, para a discussão do comportamento do tempo no país, mas essa prática requer que os meteorologistas deixem suas atividades para se dirigir a uma sala específica durante algumas horas. Os meteorologistas do SEPRE que foram consultados concordaram que um canal de comunicação permanente entre os distritos seria uma ferramenta interessante para auxiliar na interação entre os meteorologistas do Brasil inteiro e que provavelmente permitiria uma melhoria na produtividade.

Além dos meteorologistas do SEPRE, os meteorologistas que trabalham nas diversas estações climatológicas espalhadas pelo estado também consideraram importante a existência de um canal de comunicação com meteorologistas de outras estações climatológicas. Isto permitiria a troca de informações que pudessem ajudar na medição das condições meteorológicas. A principal queixa feita por um dos informantes consultados é que essa comunicação existe apenas com o NUTEL de Belém e é oficialmente restrita à transmissão da mensagem Synop.

Em resumo, um sistema para apoiar o trabalho dos meteorologistas deve:

- Oferecer facilidades para a conversão de informações meteorológicas;
- Permitir que o meteorologista use sua experiência para auxiliar o processo de previsão;
- Facilitar a comunicação informal;
- Registrar informações sobre o trabalho anterior dos diversos meteorologistas;
- Permitir maior integração com outros distritos; e
- Oferecer melhor comunicação entre as estações climatológicas.

6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Na área de CSCW estudos etnográficos são especialmente importantes para auxiliar a identificação de requisitos de aplicações para suporte ao trabalho em grupo. Simplificadamente, estes estudos permitem que certas características do trabalho possam ser identificadas assim como a motivação e importância destas características seja compreendida. Este entendimento permite identificar que características do trabalho podem e devem (ou não) ser automatizadas, ou ainda, ele sugere aspectos do sistema que devem ser mantidos flexíveis para que não atrapalhem as convenções já adotadas pelos profissionais [12].

A contribuição deste artigo é a apresentação de um estudo etnográfico visando o desenvolvimento de sistemas colaborativos para apoiar o trabalho de meteorologistas durante a elaboração de

previsões do tempo. As principais características do trabalho destes profissionais foram identificadas e traduzidas em requisitos para um sistema computacional. Em particular deve-se ressaltar o interesse destes profissionais em colaborar com seus colegas remotos. Desta forma, no momento estamos avaliando sistemas colaborativos que poderiam ser implantados para facilitar a comunicação informal entre estes profissionais.

7. AGRADECIMENTOS

Nossos agradecimentos aos meteorologistas do 2º Distrito de Meteorologia – Belém, Pará, pelo apoio dado durante a condução do estudo etnográfico apresentado neste artigo.

8. REFERÊNCIAS

[1] L. Suchman. *Plans and situated actions: the problem of human-machine communication*. Cambridge: Cambridge University Press, 1987.

[2] C. Heath and P. Luff. Collaboration and Control: Crisis Management and Multimedia Technology in London Underground Control Rooms. *Computer Supported Cooperative Work*, vol. 1, pp. 69-94, 1992.

[3] R. Bentley, J. Hughes, D. Randall, T. Rodden, P. Sawyer, D. Shapiro, and I. Sommerville. Ethnographically-informed systems design for air traffic control. *ACM Conference on Computer Supported Cooperative Work.* Toronto, Ontario, Canada, 1992.

[4] R. Grinter. Supporting Articulation Work Using Configuration Management Systems. *Computer Supported Cooperative Work*, vol. 5, pp. 447-465, 1996.

[5] R. Harper. *Inside the IMF*: Academic Press., 1998.

[6] E. M. Gerson and S. L. Star. Analyzing Due Process in the Workplace. *ACM Transactions on Office Information Systems*, vol. 4, pp. 257-270, 1986.

[7] J. E. McGrath. Methodology Matters: Doing Research in the Behavioral and Social Sciences, 1994.

[8] G. McCracken. *The Long Interview*: SAGE Publications, 1988.

[9] D. L. Jorgensen. *Participant Observation: A Methodology for Human Studies*. Thousand Oaks: SAGE publications, 1989.

[10] SYNOP Format (FM-12).

[11] Código Fonético Internacional, Wikipedia.

[12] G. Mark. Conventions and Commitments in Distributed Groups. *Computer-Supported Cooperative Work*, vol. 11, pp. 349-387, 2002.

[13] K. Schmidt. The Problem with 'Awareness' - Introductory Remarks on 'Awareness in CSCW'. *Journal of Computer Supported Cooperative Work*, vol. 11, pp. 285-298, 2002.

[14] J. Grudin. Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. *ACM Conference on Computer-Supported Cooperative Work*, Portland, Oregon, United States, 1988.

Método Colaborativo de Observação: Entendendo coletivamente os ambientes complexos

Renata Guanaes Machado, Marcos Roberto da Silva Borges, José Orlando Gomes
Programa de Pós Graduação em Informática - IM & NCE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ, Brasil
rguanaes@ufrj.br, {mborges, joseorlando}@nce.ufrj.br

RESUMO
Os métodos tradicionais de desenvolvimento de sistemas se baseiam na realização de questionários e entrevistas com usuários para a análise da situação atual que precede o levantamento de requisitos. Neste artigo, é feita uma reflexão crítica sobre estes métodos, e são apresentadas alternativas complementares que visem reduzir suas deficiências. Estas alternativas se referem à proposição de um Modelo Conceitual de Observação para a realização sistemática e estruturada de estudos naturalísticos sobre uma organização qualquer, e à elaboração de um sistema colaborativo que apóie este modelo. Este último possui o objetivo principal de auxiliar a coordenação de atividades de observação em grupo, a colaboração, interação e comunicação entre os integrantes durante as atividades de captura e análise, e a elaboração coletiva de artefatos que retratem os resultados desta observação em campo.

ABSTRACT
Traditional methods of system development have been grounded on user's interviews and questionnaires, in order to get a broad picture of the actual work environment, just before the formal requirements elicitation task. In this article, we are pondering over these methods in a critical and reflexive way, so as to recommend complementary approaches that could minimize their faults. These approaches are related to the proposition of a new "Observation Conceptual Model" that allows systematic and organized field studies about any kind of organization, including the elaboration of a collaborative system that support this conceptual model. This collaborative system has the main purpose of facilitating the coordination of observation activities in group, as well as enhancing the collaboration, interaction and communication tasks among the participants during information gathering and analyzing phases, and empowering the collective elaboration of artifacts that delineate the results of field observations.

Palavras chave
sistemas cognitivos, colaboração, etnografia, *groupware*.

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 10-18.

Keywords
cognitive systems, collaboration, ethnography, groupware.

1. INTRODUÇÃO
Em geral, os processos de desenvolvimento de sistemas [25, 28] iniciam-se com a análise da situação atual da organização, onde as pessoas relatam os problemas e explicitam as suas necessidades através de técnicas de questionários e entrevistas [12]. Em seguida, os analistas, com base nestas informações, geram os requisitos de usuário e de sistema. Entretanto, ocorrem retrabalhos nas fases subseqüentes destes processos, pois os requisitos estão falhos, incompletos e ambíguos [3, 10], e ainda, o sistema implementado pode não atender eficazmente às expectativas dos usuários [1].

Uma possível alternativa que complemente estes atuais métodos, para reduzir os problemas de qualidade das informações obtidas, é a utilização de conceitos ligados à Engenharia Cognitiva e Etnografia na atividade de análise da situação atual da organização. O enfoque principal é observar as atividades sendo executadas no próprio contexto organizacional, inclusive verificando-se os aspectos cognitivos, sociais e colaborativos embutidos nestas atividades, difíceis de serem explicitados nas entrevistas.

Este artigo apresenta a proposição de um arcabouço conceitual de análise, denominado de "Modelo Conceitual de Observação". Este modelo, de natureza sistemática e multidisciplinar, orienta a observação, captura e análise das informações, durante a fase de levantamento da situação atual de uma organização, de forma mais estruturada, completa e produtiva.

Além disso, é apresentada uma ferramenta computacional de apoio a este modelo, já que a observação deve ser realizada em equipes, principalmente no caso de atividades organizacionais complexas e distribuídas. Com isto, é possível a coordenação de atividades de observação em grupo, a colaboração e comunicação entre os integrantes, e a elaboração coletiva de artefatos, que retratem os resultados desta observação em campo, e que levem à formulação de possíveis requisitos computacionais.

Este artigo está estruturado em diferentes seções. A Seção 2 aborda resumidamente alguns métodos de análise da situação atual. Em seguida, na Seção 3 é apresentada a proposta deste artigo, que trata da elaboração de um método de observação coletiva que conta com o apoio de um *groupware*. A Seção 4 cita a metodologia de pesquisa pretendida. Finalmente, a Seção 5 resume as principais conclusões e contribuições esperadas deste trabalho.

2. REVISÃO SOBRE AS ATUAIS BASES TEÓRICAS

Esta seção detalha as fraquezas das técnicas de entrevistas e questionários, apresenta os principais conceitos das metodologias de Engenharia Cognitiva, e discute as principais características da Etnografia. Ao longo da seção, são relatados exemplos de estudos de caso e ferramentas computacionais de apoio à observação. Ao final da seção, é apresentado um quadro resumo destas abordagens.

2.1 Deficiências das Abordagens Tradicionais

Nas técnicas de questionários e entrevistas, as pessoas não relatam por completo todas as suas atividades, seja por falta de tempo, por esquecimento, ou mesmo por julgarem alguns detalhes irrelevantes.

Além disso, as pessoas podem não se dar conta ou terem dificuldades em expressar corretamente as suas habilidades, práticas e conhecimentos, relacionadas com as suas atividades, que sejam de natureza altamente tácita e intuitiva. Estas informações são difíceis de serem levantadas pelos analistas durante as entrevistas com os usuários.

Por outro lado, mesmo quando as atividades são descritas em um maior detalhamento, é possível que os analistas e usuários tenham diferentes entendimentos e interpretações [32], devido aos seus distintos modelos mentais [29].

E ainda, muitos analistas não levantam e descrevem os mecanismos cognitivos presentes nas próprias atividades, como por exemplo, as maneiras que as pessoas colaboram ou comunicam entre si, resolvem problemas ou tomam decisões em grupo, principalmente em situações críticas e imprevisíveis, difíceis de serem antecipadas. Essas informações são importantes pois, ao entender como ocorrem essas interações para a realização de uma atividade, pode-se sugerir e projetar requisitos que apóiem essas atividades colaborativas.

Em organizações complexas e de grande porte, é normal que se estabeleça uma divisão de tarefas de análise entre os analistas, o que possibilita a ocorrência de divergências ou diferentes pontos de vista sobre as informações coletadas pela equipe. Como conseqüência, são necessários esforços adicionais para confirmação, conciliação e validação dos dados capturados, muitas vezes voltando-se novamente aos usuários.

Em resumo, as deficiências relatadas não permitem que sejam captadas as descrições completas e reais das atividades executadas e de seus mecanismos de cognição e colaboração. Portanto, torna-se difícil a compreensão dos problemas existentes, das dificuldades enfrentadas pelas pessoas durante a realização dessas atividades e das suas efetivas necessidades de informatização e automação, seja completa ou parcial.

2.2 Valorização das Abordagens Cognitivas

Fatores como o crescente uso da tecnologia de informação, aumento da automação e a especialização do trabalho, resultam na atual caracterização das organizações como complexas, dinâmicas e distribuídas, e no reconhecimento e aumento das tarefas ditas cognitivas.

Desta forma, a execução de atividades organizacionais, sejam operacionais ou gerenciais, não envolve somente aspectos físicos ou cognitivos, e sim ambos [4], sendo possível a simultaneidade de ações e decisões [27], embora a cognição seja cada vez mais predominante.

As abordagens cognitivas capturam e analisam uma diversidade de tópicos, como processos mentais ou de raciocínio, percepção, interpretação, memória e tomada de decisão, que se encontram nas atividades de planejamento, execução, elaboração de diagnósticos e resolução de problemas. Os aspectos sociais, tais como relacionamentos, colaboração, cooperação e comunicação são também considerados importantes aspectos cognitivos [21], inclusive o contexto, que influencia a cognição e performance humana [31].

Em virtude da possibilidade de se investigar as diversas variáveis de cognição, a literatura apresenta uma abrangente coleção de modelos conceituais e métodos para a captura, análise e avaliação de dados coletados durante uma investigação, cada qual para uma determinada finalidade. A seleção de quais métodos empregar depende dos objetivos do estudo e de quais aspectos cognitivos que são mais relevantes. Além disso, há um rico relato de aplicação destes por meio de estudos de caso em campo [16].

São discutidos a seguir dois métodos cognitivos, pois a partir destes é proposta a alternativa complementar de se analisar situação atual de um ambiente de trabalho.

2.2.1 Análise do Trabalho Cognitivo (Cognitive Work Analysis, CWA)

Essa metodologia se apresenta como uma abordagem genérica e holística para análise e investigação de ambientes de trabalho e das suas respectivas adaptações frente às complexidades [9], a fim de gerar requerimentos de sistemas computacionais.

De acordo com a Figura 1, a metodologia se apresenta em diferentes níveis de abstração. Estes níveis de abstração diferem do paradigma de "dividir para conquistar", pois não se quebra o problema em diferentes partes, mas analisa-se a organização em diferentes perspectivas: o atual ambiental de trabalho, organizacional, domínio, atividade e individual (Análise usuários e Análise ergonômica). Níveis mais baixos são influenciados pelas características dos níveis mais altos.

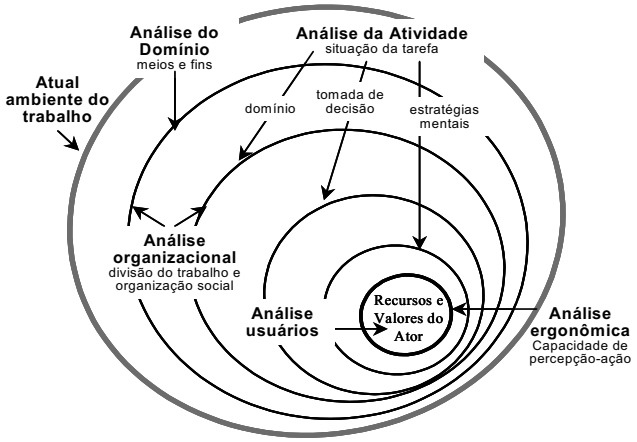


Figura 1: Dimensões de Análise (Adaptado de [9])

Por exemplo, no nível ambiental, pode-se analisar as variáveis externas à organização, como regulamentações; no nível organizacional, o estilo de gestão e cultura organizacional existente; no nível de domínio são verificados os objetivos e os recursos humanos e tecnológicos disponíveis para a execução

destes; enquanto que no nível da atividade é realizado o estudo mais detalhado da interação humana -informação, tais como estratégias e métodos para a execução das tarefas e decisões.

Estes níveis não são obrigatórios e fixos, mas escolhidos e definidos conforme o escopo da investigação. Nestes níveis, o foco não é a identificação e descrição de variáveis situacionais que afetam a interação humana -informação no nível mais baixo, mas sim a descoberta e reflexão dos fatores que delimitam, formam e influenciam esta interação.

Com isso, avalia-se o sistema sócio-técnico atual no ambiente de trabalho, para que em seguida sejam geradas recomendações para novos projetos de sistemas, conforme o contexto estudado. Esta abordagem difere do típico ciclo de engenharia de software, onde o sistema é concebido, modificado e testado diversas vezes até que seja aceito pelo usuário na fase de homologação.

Além disso, trata-se de uma abordagem centrada no trabalho e na atividade. As pessoas que interagem com sistema sócio-técnico são denominadas de atores envolvidos em suas ações relacionadas com o trabalho, e não simplesmente como usuários de sistemas. a Consideram-se certos atributos destes como típicos e característicos, sendo possível a formulação de um estereótipo.

A abordagem possui enfoque formativo, ao invés de descritivo ou normativo. Para a execução de uma tarefa, o enfoque descritivo cita o procedimento mais comum; o normativo constitui o melhor modo conforme critérios pré-estabelecidos; enquanto que o formativo apresenta alternativas viáveis, que serão escolhidas conforme o contexto da situação e perfil de um usuário.

Desta forma, se for utilizado o enfoque descritivo ou normativo para a análise da atividade, percebe-se que os sistemas projetados não serão capazes de apoiar a adaptação humana às situações inesperadas ou não antecipadas pelos desenvolvedores. Ou seja, não basta somente descrever as atividades, é preciso predizer o comportamento humano sob diversas circunstâncias, para disponibilizar estas alternativas no sistema.

Como um exemplo da aplicação prática desta metodologia, pode-se citar a melhora de sistemas de controle de processo térmico-hidráulico [15], por meio da análise das dimensões do domínio e da atividade que levaram aos requisitos de interfaces e de automação de ações e decisões.

Entretanto, esta metodologia requer a realização de intensivos estudos em campo e atuação de especialistas; não apresenta qualquer mecanismo de auxílio à sua execução e prática, seja na forma de guias, seja através de ferramentas computacionais de apoio. E o mais importante, não apresenta um processo sistemático e estruturado para se transformar os resultados da análise em requisitos formais de sistemas

2.2.2 Análise Aplicada do Trabalho Cognitivo (Applied Cognitive Work Analysis, ACWA)

Trata-se de uma metodologia sistemática e iterativa, com o objetivo principal de se levantar os requisitos para sistemas de apoio à decisão [8]. Desta forma, foca-se nas dimensões de domínio e atividade, pois são as mais relevantes para apoiar a tomada de decisão.

Propõe um processo estruturado, que se inicia com a análise do domínio e finaliza com a geração de uma especificação funcional para a equipe de desenvolvimento, onde os resultados da análise são transformados sistematicamente para requisitos de informações e interfaces.

A metodologia é segmentada em 5 etapas, conforme a Figura 2, iniciando-se com a representação formal do domínio em uma Rede de Abstração Funcional [26], semelhante a um modelo mental de um especialista. Esta rede explicita os objetivos a serem alcançados e os meios funcionais disponíveis para os mesmos.

Este domínio, uma vez implementado em um sistema, faz com que este apresenta as informações necessárias, já filtradas e tratadas, ao invés de disponibilizar dados que seriam ainda interpretados e processados pelos usuários. Desta forma, o sistema intuitivamente se adapta aos seus processos cognitivos, complementando e apoiando a tomada de decisão humana.

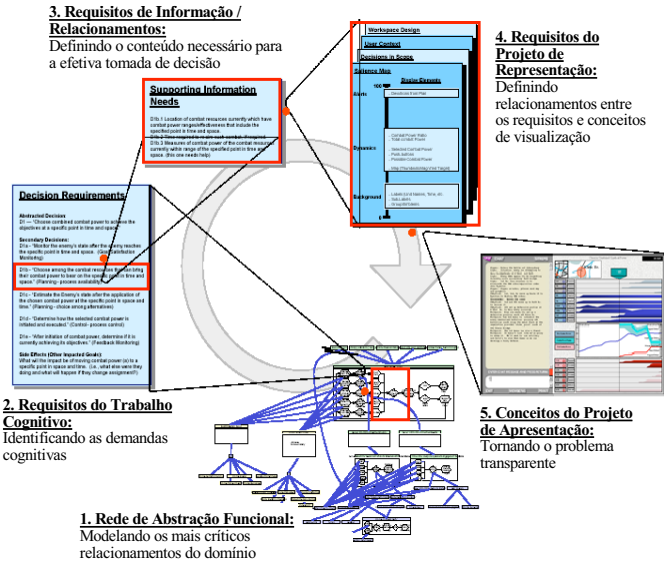


Figura 2: Processo de Análise (Adaptado de [8])

Desta rede parte-se para as demais etapas, representadas na Figura 2. Na segunda etapa, derivam-se os requisitos de cognição que caracterizam as atividades de reconhecimento, raciocínio e decisão para resolução de problemas. Em seguida, na terceira etapa, são identificadas as informações necessárias para a resolução de cada requisito de cognição. Finalmente, nas próximas duas etapas, são definidos os requisitos de projeto, ou seja, como estas informações devem ser representadas e implementadas em um sistema de informação.

Como um exemplo da aplicação prática desta metodologia, pode-se citar a análise do ambiente de trabalho de investigadores militares, que resultaram na elaboração de um protótipo computacional, com interfaces inteligentes e intuitivas [23]. O objetivo deste é apoiar as operações de identificação de possíveis ataques terroristas, antecipadamente, a fim de garantir a segurança nacional.

Entretanto, esta metodologia não especifica como realizar a elicitação dos conhecimentos do domínio e os estudos em campo, bem como não utiliza ferramentas computacionais de apoio à execução da mesma.

2.3 Importância da Abordagem Etnográfica

A abordagem etnográfica é uma metodologia oriunda da Antropologia Social, onde se procura entender e descrever uma

nação, povo ou cultura, até então desconhecida, por meio da observação natural. Nestes estudos, inclui-se a investigação e análise dos comportamentos, rituais, técnicas, saberes e práticas pertencentes a essas culturas. É o caso de missionários que procuraram entender os povos indígenas para em seguida educá-los e catequizá-los, por exemplo.

A etnografia pode ser realizada de diferentes formas além da observação em campo propriamente dita, como a execução de variados tipos de experimentos controlados e simulações; uso de protocolos verbais; uso de equipamentos de áudio, vídeo e fotografias; participação ativa; e as mais variadas categorias de entrevistas, como estruturadas e não estruturadas.

Suchman [30] foi a primeira pesquisadora a propor a realização da abordagem etnográfica em projetos de tecnologias e sistemas, levando-se em consideração os aspectos sociais e contexto de uso. Em seguida, muitos outros pesquisadores a adotaram para estudo e análise dos mais variados domínios de atividades, como controle de tráfego aéreo, metrô e usinas nucleares, ambiente de escritórios, entre muitos outros.

Desta forma, a adoção da Etnografia é uma alternativa viável de se informar requisitos de sistemas [17], mais especificamente, sistemas interativos e de apoio ao trabalho em grupo.

Os analistas observam as atividades sendo executadas no contexto específico, o que possibilita: a descoberta dos aspectos sociais, cognitivos e colaborativos, a percepção de como ocorre a interação com os artefatos existentes, e autêntica identificação das dificuldades e adaptações das pessoas frente aos problemas.

Uma outra potencial contribuição é a melhora da comunicação entre os usuários e a equipe de analistas e desenvolvedores, já que a atuação em campo e o maior contato permitem o acesso à cultura, ao modo real do trabalho das pessoas e ao entendimento da perspectiva e de seus pontos de vista.

De acordo com os resultados de uma pesquisa [6], as ferramentas de observação direta, como gravador, vídeo e câmeras digitais, são amplamente utilizadas em campo, com percentual de uso de 85% a 98%. A utilização de softwares de apoio chega a 50%, embora focados na observação de fatores fisiológicos e ambientais, tais como postura e nível de iluminação. Estes softwares não tratam dos aspectos cognitivos, sendo um indicativo da contribuição potencial da proposta deste artigo, que é a elaboração de ferramenta computacional de apoio à discussão e análise, em grupo, dos fatos observados e registrados.

O software RATE (Remote Analysis of Team Environment) é um exemplo do uso conjunto de sistemas e equipamentos de áudio e vídeo como apoio à observação remota de equipes de médicos e enfermeiras, durante as cirurgias, com o objetivo de analisar e avaliar os seus aspectos comportamentais e de performance [13].

As imagens de vídeo, uma vez capturadas, são categorizadas, classificadas e analisadas, durante a própria observação, conforme a Figura 3. A categorização é feita conforme o tipo de atividade ou interação ocorrida, como por exemplo, é uma "solicitação de informação" ou trata-se de "comunicação de um problema". Já a classificação se baseia na qualidade da atividade, se realizada com sucesso ou se com a ocorrência de um erro, por exemplo.

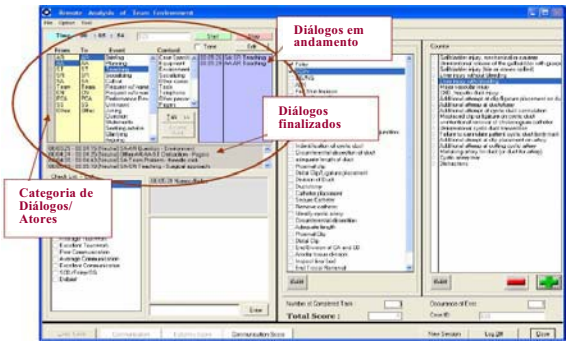


Figura 3: Categorização das interações no RATE [13]

Sua utilização é possível em quaisquer domínios, pois a aplicação é flexível e configurável. Desta forma, esta ferramenta pode ser adaptada para propósitos de informar requisitos de sistemas de informação.

Entretanto, a etnografia em geral se apresenta como uma metodologia não sistemática e cara, pois demanda um longo tempo de captura e a presença de vários observadores ou especialistas. Além disso, apresenta os resultados da investigação no formato discursivo, descritivo e textual, que não atende aos adjetivos de formalismo e objetividade requeridos pelos desenvolvedores de sistemas [18].

Como um esforço de organizar e ressaltar os dados mais relevantes para as equipes envolvidas, cada qual interessada em certos atributos do ambiente de trabalho, foi criada a ferramenta computacional Designer´s Notepad [18]. Em resumo, os dados são apresentados em hipertexto, ocasionando uma navegação focada no que é considerado importante e possibilitando a abstração das informações necessárias.

A ferramenta possui como principais elementos: componentes gráficos, referências cruzadas e notas de texto. Estes elementos, em conjunto, permitem que o material etnográfico seja estruturado em diferentes formas de representação e perspectivas, como por exemplo, apresentando o processo de trabalho formal ou layout físico de um local de trabalho, conforme a Figura 4. As referências cruzadas indicam vínculos de entidades para outros desenhos, e notas de texto são narrativas textuais complementares. Notas acrescentadas a uma pessoa podem representar transcrições de entrevistas ou relato de suas reais atividades e ações.

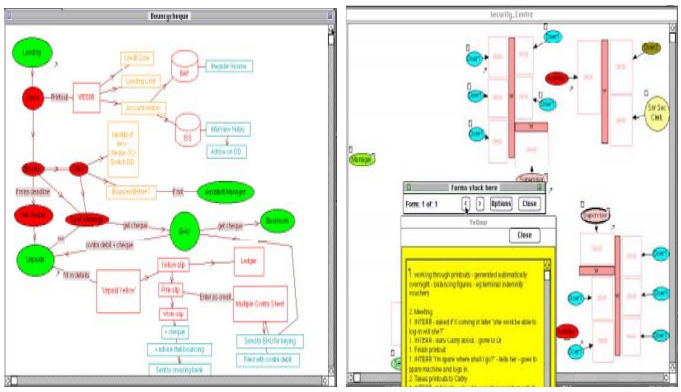


Figura 4. Perspectivas de Visualização da Etnografia

Tabela 1. Visão Geral das Abordagens de Análise da Situação Atual

Abordagem	Principais Características	Vantagens	Desvantagens
Entrevistas e Questionários	Reuniões entre analistas e usuários para levantar problemas e necessidades.	Métodos simples, diretos, rápidos e fáceis de serem executados.	Informações falhas, incompletas e ambíguas; Aspectos cognitivos e sociais não capturados.
Análise do Trabalho Cognitivo	Análise de ambientes de trabalho através de níveis de abstração e enfoque formativo.	Abordagem flexível; descoberta dos fatores que delimitam e influenciam a interação humana - informação.	Requer intensos estudos em campo; ausência de apoio e processo para geração de requisitos de sistemas.
Análise Aplicada do Trabalho Cognitivo	Análise do domínio e da atividade para especificação do processo de tomada de decisão humana.	Apresenta processo estruturado desde análise do domínio até requisitos de sistemas; orienta como gerar artefatos em cada etapa do processo.	Não especifica como elicitar o domínio; não orienta como realizar os estudos em campo.
Etnografia	Observação natural dos ambientes de trabalho.	Descoberta dos aspectos sociais, cognitivos e colaborativos; informações coletadas são autênticas.	Técnica não sistemática, de longa duração e cara; exige vários observadores; formato textual.

Porém, não são mencionadas as atividades de coordenação e comunicação entre os observadores durante a etnografia, ou seja, de que forma estes chegaram aos resultados.

Por outro lado, a fim de atender às demandas limitadas de tempo e recursos, foram propostas diferentes abordagens etnográficas para informar requisitos de sistemas [17], além da etnografia ágil [22]. Esta agiliza a observação através de delimitação do foco; seleção de pessoas chave; múltiplas observações ou observações interativas; e uso de métodos analíticos colaborativos, como elaboração de mapas conceituais e contagem de histórias. O objetivo não é a definição completa do domínio, mas somente a identificação dos aspectos de sociabilidade no ambiente de trabalho, e de que forma estes afetam a eficiência das tarefas.

3. PROPOSTA DE TRABALHO

São apresentados, nas seções anteriores, breves relatos explicativos, incluindo estudos de caso e exemplos de ferramentas, sobre metodologias cognitivas e técnicas etnográficas, todas predominantemente focadas no objetivo principal de se levantar requisitos de sistemas.

Com isso, o objetivo desta seção é mostrar a utilização de um conjunto de conceitos dos métodos apresentados, acrescentando-se demais idéias da área de trabalho colaborativo [2, 19], para a proposição de uma alternativa complementar para a execução da atividade de análise da situação atual.

Enfatiza-se que a proposta de trabalho busca a complementaridade dos atuais métodos de entrevistas e questionários, e não a sua substituição por esta.

3.1 Modelo Conceitual de Observação

Como forma de superar os problemas de falha, incompleteza e ambigüidade das informações obtidas em entrevistas e questionários, a proposta deste trabalho se baseia na elaboração do arcabouço teórico denominado de “Modelo Conceitual de Observação”.

Este modelo tem a função geral de apoiar a etnografia em grupo, auxiliando e orientando os analistas a observar, descrever e analisar certos ambientes de trabalho, de natureza complexa e distribuída, de forma colaborativa, estruturada e sistemática.

Conforme a Figura 5, este modelo é composto de várias dimensões de análise, criadas e adaptadas a partir da metodologia CWA [9], bem como tem um processo de transformação destas dimensões para a geração de artefatos e requisitos, criado e adaptado a partir da metodologia ACWA [8].

A partir das Dimensões de Análise, o modelo especifica "o que observar" e "como observar". O termo "o que observar" se refere às variáveis do ambiente de trabalho que devem ser capturadas em campo, principalmente aquelas relacionadas com os aspectos de cognição. Já o termo "como observar" menciona as formas possíveis de coletar e registrar as observações, que podem ser através do uso de equipamentos de áudio e vídeo, múltiplos observadores no próprio local, cada qual com apoio de artefatos tecnológicos ou não, como notas em campo, padrões de observação e computadores portáteis. Os padrões de observação são formulários prontos onde os observadores checam, verificam e registram anotações, sendo uma opção mais eficiente à transcrição manual de eventos ou ações.

Múltiplos observadores podem ser utilizados de forma que cada analista investiga uma diferente perspectiva, ou então, que mais de um analista capture as mesmas variáveis.

Nem todas as dimensões serão utilizadas em uma etnografia. Com isso, a decisão de quais dimensões aplicar depende do escopo e da finalidade da análise, que por si geram diferentes tipos de artefatos. Por exemplo, no caso de salas de controle de usinas nucleares, o escopo pode ser a análise de uma atividade específica, como o controle dos rejeitos nucleares, com a finalidade de se avaliar a interação dos operadores com as interfaces representativas dos processos de separação e tratamento dos rejeitos. Neste caso, uma dimensão de análise relevante é aquela em que se verificam os mecanismos de percepção: como operadores detectam um problema e se as interfaces apresentam informações corretas; ou de tomada de decisão: em que condições estes seguem procedimentos formais ou recorrem às experiências anteriores para resolver problemas críticos e imprevisíveis.

Além disso, voltando-se na Figura 5, a partir dos registros destas observações, armazenados nas bases de informação, o modelo também descreve "como analisar", direcionando os esforços do grupo para a elaboração coletiva de levantamentos preliminares, diagnósticos, ou demais artefatos, de tais ambientes observados, por meio de processos colaborativos de exposição de idéias, argumentação, reflexão e negociação de possíveis divergências ou conflitos de informações.

Sempre que possível ou necessário, pode-se voltar ao campo para um contínuo refinamento, confirmação das coletas de dados iniciais ou para maior aprofundamento de variáveis de análise. Desta forma, a atividade de observação pode ocorrer em todas as fases de desenvolvimento de um sistema, e não deve limitar-se somente à etapa de captura ou levantamento inicial.

Com isso, o grupo de analistas tem a percepção tanto da criação como da evolução dos artefatos, de forma a discutir, validar e complementar com demais informações. A grande diferença, deste modelo para os métodos de entrevistas, é que os analistas terão acesso às informações, coletadas por outros, durante a elaboração das mesmas, e não quando estas estiverem já finalizadas em documentos completos.

O papel do Processo de Análise é orientar a criação de sucessivos artefatos, ou seja, quais documentos devem ser produzidos até a geração dos requisitos. Os processos colaborativos, mencionados anteriormente, ocorrem repetidamente para cada etapa do Processo de Análise.

Finalmente, ainda se referindo à Figura 5, após o término das atividades colaborativas de análise, o modelo apresenta opções de "como gerar artefatos". Este termo se refere à apresentação das informações coletadas e analisadas pelo grupo, sob diferentes vistas ou perspectivas, em interfaces computacionais. A navegação entre as interfaces permite o acesso às informações mais relevantes para um determinado usuário, bem como indica a rastreabilidade dos artefatos gerados para os requisitos de informação ou de automação.

As diferentes perspectivas, mencionadas na ferramenta Designer´s Notepad [18] como formas de representação dos resultados da observação, são adaptadas e consideradas como produtos gerados da análise colaborativa do grupo de analistas. Cada Dimensão de Análise gera um determinado artefato. Por exemplo, os analistas utilizam a dimensão do domínio da organização para a elaboração, em grupo, da perspectiva denominada de "Modelo Mental do Domínio".

Nota-se, na Figura 5, que ambas as atividades de "como analisar" e "como gerar artefatos" contam com o apoio de uma ferramenta computacional de apoio ao trabalho cooperativo, ou *groupware* [19]. Com isso, de forma a propiciar a colaboração, participação e análise coletiva, o modelo tem também a descrição de possíveis papéis e responsabilidades para cada analista do grupo de observadores, que são detalhados na seção seguinte deste artigo, que trata das principais funcionalidades da ferramenta.

Dois importantes pressupostos são seguidos neste modelo de

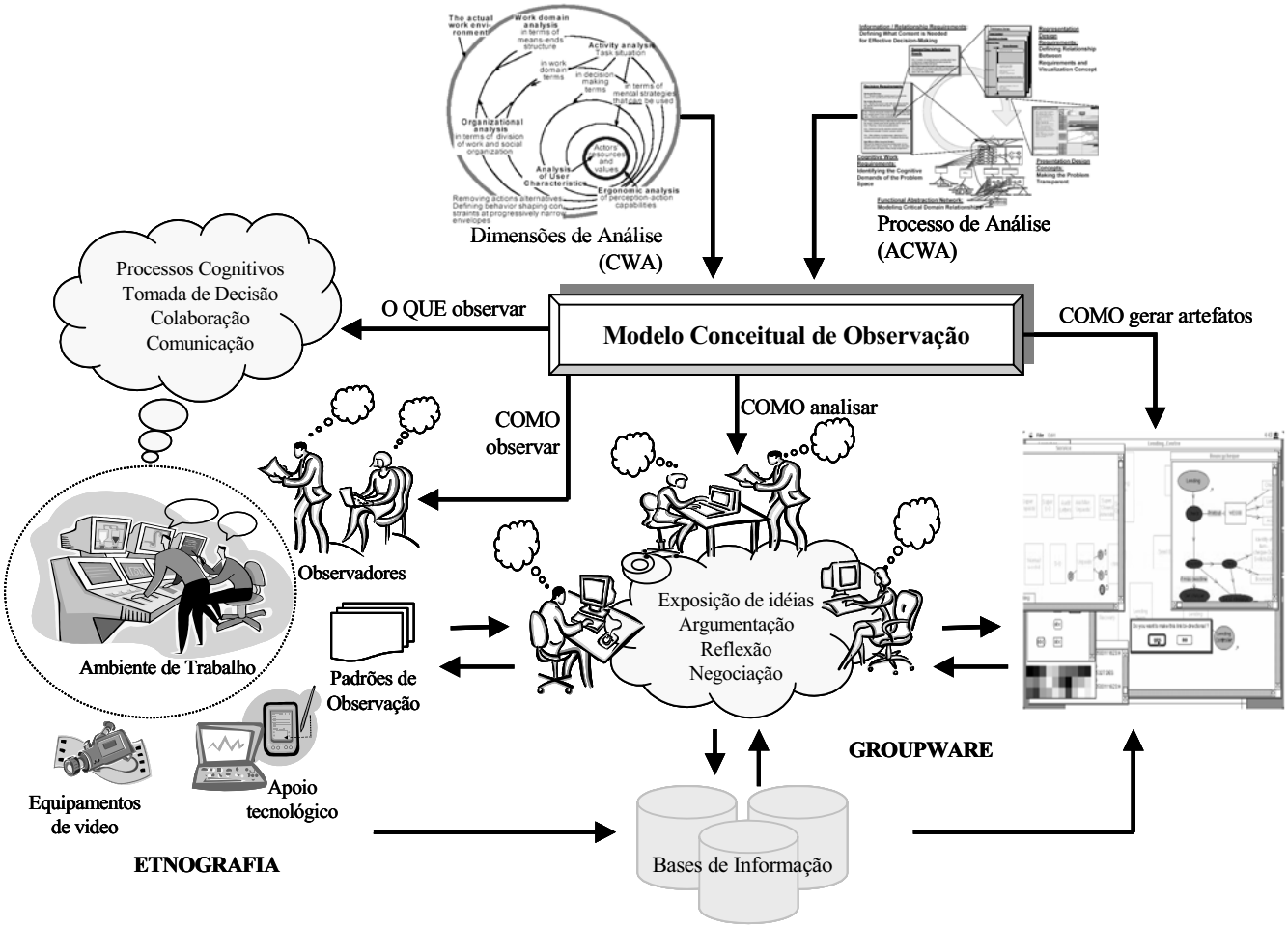


Figura 5. Premissas do Modelo de Observação

observação. O primeiro é a participação das pessoas da própria organização, ou seja, do ambiente analisado, que estão sendo observadas, a fim de que estas possam discutir e validar as interpretações sobre as atividades, bem como compartilhar demais conhecimentos ainda não explicitados. Os analistas podem ser os mais experientes observadores, mas somente as pessoas da organização contêm os conhecimentos vitais das suas próprias atividades, sendo sua participação de extrema importância na análise da situação atual.

O segundo é que a atividade de análise da situação atual ocorrerá em organizações que já possuam algum nível de informatização, ou seja, sistemas já implementados mas que necessitam de melhorias ou de total substituição por novos sistemas. Com isso, as atuais atividades executadas não irão sofrer alterações radicais. Caso contrário, poderá ser desnecessário esse levantamento da análise da situação atual no ciclo de desenvolvimento de um novo sistema, já que este mudaria todo o processo de trabalho.

3.2 Ferramenta Computacional de Apoio ao Modelo Conceitual

Com o aumento da complexidade das tarefas organizacionais, que resulta na especialização do trabalho, torna-se mais clara a evidência de que as pessoas não trabalham sozinhas, e sim em equipes de trabalho multidisciplinares, e ainda, dispersas geograficamente [20]. Desta forma, além da necessidade de se observar estas equipes, a própria atividade de observação deve ser realizada por um grupo de analistas.

Com isso, a criação de uma ferramenta de apoio ao trabalho em grupo, denominada de *groupware* [19], possibilita uma maior eficiência na execução do método de observação, que já se pressupõe ser de natureza coletiva. O foco é o uso do computador não como meio de solução de problemas, mas como meio de facilitar as interações humanas [7]. Desta forma, a ferramenta, tem a meta estratégica de minimizar as dificuldades existentes nas atividades em grupo que são realizadas face-a-face, bem como de otimizar custos, caso as observações sejam desempenhadas de forma distribuída, ou a análise destas sejam feitas por pessoas de diferentes locais.

Além dessa meta, os principais objetivos da ferramenta são: gerar um ambiente de trabalho colaborativo, e mesmo de aprendizagem, já que haverá compartilhamento de conhecimentos e experiências; possibilitar a coordenação de atividades de observação em grupo; fomentar a colaboração e comunicação entre os integrantes; e apoiar a elaboração coletiva de artefatos que retratem os resultados desta observação em campo.

A fim de possibilitar a criação de um ambiente colaborativo, cada novo usuário na ferramenta terá a possibilidade de colocar uma descrição de seu perfil, como o cadastro de dados pessoais, a sua formação acadêmica, experiências em projetos e conhecimentos nos campos da engenharia de software, ergonomia, engenharia cognitiva e etnografia, por exemplo. O objetivo do detalhamento deste perfil é propiciar um ambiente de equipe, camaradagem e confiança uns nos outros [24].

Já a coordenação de atividades é essencial para que se possa planejar as tarefas individuais de cada analista para a efetiva observação, e verificar posteriormente o registro de dados e a análise dos resultados desta observação. Sem esta coordenação, há o risco de ocorrerem conflitos e divergências sobre o levantamento da situação atual por diferentes analistas, bem como destes realizarem tarefas repetitivas. [5]. A gestão das tarefas e a negociação de conflitos são primordiais para que sejam obtidos o planejamento eficiente da atividade de observação, o equilíbrio nos diferentes pontos de vista e o consenso coletivo.

Como forma de permitir tal mecanismo de coordenação na ferramenta, são criados diferentes perfis de usuários, como por exemplo: administrador, moderador, editor e usuário, entre outros. A ferramenta permite a criação de demais tipos de usuários, se necessário.

O perfil de administrador tem a permissão de implementar novas técnicas e ferramentas de análise em uma base de métodos, como os padrões de observação mencionados anteriormente. A ferramenta deve ser flexível e configurável, de forma a permitir a adaptação e alteração destes métodos conforme o ambiente a ser analisado.

O perfil de moderador, semelhante a um gerente de projeto e adequado para uma pessoa responsável pelo grupo de analistas, tem vários objetivos: estabelecer as dimensões de análise, escolher as técnicas e ferramentas de observação disponíveis na base de métodos, definir as tarefas e os papéis de cada analista, propor prazos para a geração de artefatos, conduzir a dinâmica das interações, determinar discussões, gerenciar conflitos de idéias, formar votações, entre outras inúmeras ações.

Ainda para o perfil de moderador, a ferramenta apresenta um resumo das interações do grupo, como os artefatos inicialmente criados por cada membro, a quantidade de alterações e a quantidade de contribuições por artefato, as tarefas cumpridas ou atrasadas conforme os prazos estipulados, entre outras informações. Com isto, é possível ao moderador identificar o andamento geral da análise, cobrar pelas tarefas atrasadas, motivar aqueles que não contribuíram, além de demais atividades de monitoramento e controle.

O perfil de editor tem a responsabilidade de criar uma versão final de um artefato após a execução de uma análise colaborativa pelo grupo de analistas.

Ao perfil de usuário, recomendado para aqueles que serão observados, é permitida a leitura dos artefatos em andamento, postagem de avaliações e comentários.

Em comum, todos os perfis têm a permissão de propor, defender ou contestar conceitos ou idéias, além de revisar e validar artefatos finalizados pelo perfil de editor.

Na medida do possível, a ferramenta tem a tarefa de inserir os dados de contexto para cada artefato, como por exemplo: o nome do usuário, ações ou alterações realizadas no artefato, data e hora da sua contribuição. Com isso, é possível o acompanhamento da criação e evolução coletiva do artefato pelos analistas, sendo fundamental que cada um receba o retorno dos outros sobre o seu próprio trabalho ou contribuição individual [11].

Percepção é o conhecimento a respeito das ações e interações de outros membros em um espaço compartilhado de trabalho [14]. A ferramenta implementa a percepção no ambiente de trabalho colaborativo, de forma a informar as pessoas que estão logadas; os artefatos que estas estão trabalhando no momento, bem como as suas atuais tarefas e atribuições; os novos comentários em cores diferentes para aqueles usuários que ainda não os visualizaram, entre outras. Com isso, a ferramenta representa mecanismos de se notar como anda a atividade do grupo,

atentando-se para evitar a distração dos usuários individuais com estas representações.

Com relação aos estímulos à comunicação, primordiais para um trabalho colaborativo, a ferramenta permite a adaptação do tempo e espaço do grupo de analistas, de forma a admitir a colaboração síncrona e assíncrona, bem como remota e distribuída, dos seus membros. São implementadas diversas facilidades de comunicação do grupo, como e-mails, fóruns de discussão, chat e quadro de avisos.

Em resumo, a ferramenta implementa efetivamente os mecanismos de cooperação, colaboração, comunicação e percepção, características próprias de sistemas de *groupware* [19]. E ainda, esta ferramenta, uma vez alinhada com o Modelo Conceitual de Observação, apóia a geração, coordenação, negociação e compartilhamento dos conhecimentos resultantes da tarefa de observação de um ambiente de trabalho.

O produto, resultado da aplicação do método e do uso da ferramenta, é um conjunto de informações gerado a partir de conhecimentos, idéias e percepções de um grupo formado pelos analistas e pelas pessoas da organização, servindo de entrada para o início da proposição de um novo sistema.

4. METODOLOGIA DE PESQUISA

Após a elaboração dos produtos deste trabalho, ou seja, do modelo conceitual e da ferramenta computacional, ambos serão testados e avaliados em estudos de caso.

O objetivo principal destes estudos é a comprovação da aplicabilidade desta alternativa de análise da situação atual, isto é, se esta efetiva a geração completa e menos ambíguas de requisitos de sistemas, além da verificação da eficácia do próprio trabalho em grupo, por meio da ferramenta que implementa os conceitos de *groupware*.

Nós estamos delimitando a elaboração do modelo conceitual e da ferramenta para aplicação em ambientes de trabalho onde existe a interação de pessoas com interfaces computacionais de monitoramento, comando e controle. Exemplos deste ambiente são: salas de controle de processos industriais, usinas nucleares, controle de tráfego aéreo, e até mesmo ambientes onde se monitoram o andamento de ações na bolsa de valores.

As salas de controle são caracterizadas por uma variedade de interfaces, painéis, comandos, controles, sensores e alarmes, além de soluções tecnológicas e de engenharia. As ações dos operadores nestes ambientes são normalmente baseadas em procedimentos operacionais previamente estabelecidos e planos de segurança. Entretanto, acidentes acontecem devido às falhas humanas decorrentes de sistemas inadequados, como Three Miles Island, nos EUA, e Chernobyl, na Ucrânia.

O uso de etnografia nestes ambientes se justifica pela necessidade de se levantar e investigar os meios pelos quais as pessoas colaboram entre si e se adaptam a situações críticas e imprevistas, onde os procedimentos não existem, são ineficientes ou parcialmente aplicados. Estes eventos são de natureza tácita e emergente, ou seja, não são conhecidos previamente, não são registrados e são dificilmente reportados em entrevistas.

Com isso, nós acreditamos que essa nova alternativa de análise trará importantes conhecimentos que serão aplicados para a construção de novos sistemas ou melhora de sistemas existentes, que sejam de natureza mais adaptativa e complementar às capacitações humanas.

5. CONCLUSÃO

Frente à necessidade de se analisar ambientes de trabalho que cada vez mais se apresentam como complexos, dinâmicos e distribuídos, ou ainda, compostos de inúmeras variáveis, nós acreditamos que as abordagens cognitivas e etnográficas possam trazer importantes resultados, na forma de requisitos mais completos e menos ambíguos, que serão úteis para os futuros projetos de sistemas.

Além disso, a tarefa de observação, uma vez realizada por um grupo de analistas com diferentes papéis e responsabilidades, incluindo a participação das pessoas sendo observadas, ocasiona a geração colaborativa de artefatos, que representem o conjunto de contribuições e idéias de diferentes observadores e demais integrantes.

Para isso, este trabalho de pesquisa se baseia na proposição de um método colaborativo apoiado por uma ferramenta computacional, de forma a especificar como coordenar e realizar a observação em equipe, bem como proporcionando o fomento à colaboração e à comunicação durante a atividade de captura e análise.

Este trabalho traz uma contribuição para o atual campo de pesquisa, pois conforme observado na literatura, a observação, apoiada por ferramentas ou não, embora procure analisar as atividades e a performance de equipes de trabalho, não se apresenta como uma atividade colaborativa dos próprios observadores.

Até o momento da escrita deste artigo, estamos atualmente em processo de elaboração, detalhamento e refinamento do modelo, para que em seguida possamos criar um protótipo computacional e testá-lo em estudos de campo.

6. REFERÊNCIAS

[1] Ackoff, R.L. *Management Mis-Information Systems*. Management Science, 14, 1967.

[2] Bannon, L.J., and Schmidt, K. Four characters in search of a context. In: Bowers, J. M., and Benford, S. D. (Eds.), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*. Amsterdan, North Holland, 1991.

[3] Blaschek, J.R. *Gerência de Requisitos, o principal problema dos projetos de software*. Rio de Janeiro, 2003. Disponível em http://www.bfpug.com.br/islig-rio/Downloads/Ger%C3%AAncia%20de%20Requisitos-o%20Principal%20Problema%20dos%20Projetos%20de%20SW.pdf. Acesso em Junho 2006.

[4] Clark, A. *Being there: Putting brain, body and world together again*. MIT Press, London, 1997.

[5] Cruz, A. J. A. da, Raposo, A. B., e Magalhães, L.P. Coordination in Collaborative Environments -A Global Approach. In: *Proceedings of the 7th International Conference on CSCWD 2002*. Rio de Janeiro, 2002, 25-30.

[6] Dempsey, P.G., McGorry, R.W., and Maynard, W.S. A survey of tools and methods used by certified professional ergonomists. *Applied Ergonomics*, 36, 4 (July 2005), 489-503.

[7] Ellis, C.A., Gibbs, S.J., and Rein, G.L. Groupware: some issues and experiences. *Communications of the ACM*, 34, 1, January 1991, 38-58.

[8] Elm, W.C., Potter, S.S., Gualtieri, J.W., Roth, E.M., and Easter, J.R. Applied cognitive work analysis: a pragmatic methodology for designing revolutionary cognitive affordances. In: Hollnagel, E. (Ed.), *Handbook for Cognitive Task Design.* Lawrence Erlbaum Associates, London, 2003, 357-382.

[9] Fidel, R., and Pejtersen, A.M. From information behavior research to the design of information systems: the Cognitive Work Analysis framework. *Information Research*, 10, 1 (October, 2004), paper 210. [Available at http://InformationR.net/ir/10-1/paper210.html]

[10] Freitas, D.P. *Ampliando a colaboração no levantamento de requisitos de sistemas*. Dissertação (Mestrado em Informática). Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

[11] Fucks, H., and Assis, R. Facilitating perception on virtual learning based environments. *Journal of Systems and Information Technologies,* Vol. 5, no. 1, Edith Cowan University, 2001, 93-113.

[12] Goguen, J.A., and Linde, C. Techniques for requirements elicitation. In: *IEEE International Symposium on Requirements Engineering*, 1, San Diego, 1993, 152-164.

[13] Guerlain, S., Shin, T., Guo, H., Adams, R., and Calland, J.F. The RATE tool: multimedia observation and analysis of teams. In: *Proceedings of the 46th Annual Meeting of the Human Factors and Ergonomics Society*, Santa Monica, CA:HFES, 755.

[14] Gutwin, C., Roseman, M., and Greenberg, S. A usability study of awareness widgets in a shared workspace groupware system. In: *Proceedings of CSCW'96*, Boston, Massachusetts, November 1996.

[15] Hajdukiewicz, J.R., and Vicente, K.J. A theoretical note on the relationship between work domain analysis and task analysis. *Theor. Issues in Ergon. Sci*., 5, 6 (December 2004), 527-538.

[16] Hollnagel, E. *Handbook for Cognitive Task Design*. Lawrence Erlbaum Associates, London, 2003.

[17] Hughes, J.A., King, V., Rodden, T., and Andersen, H. Moving out from the control room: ethnography in system design. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, Chapel Hill, North Carolina (October 1994), 429-439.

[18] Hughes, J.A., O'Brien, J., Rodden, T., and Rouncefield, M. *Ethnography, Communication and Support for Design*. Technical Report Ref. CSEG/24/1997, Lancaster University, Lancaster, UK, 1997. [Available at http://www.comp.lancs.ac.uk/computing/research/cseg/97_rep.html]

[19] Khoshafian, S., and Buckiewicz, M. *Introduction to Groupware, Workflow, and Workgroup Computing*. John Wiley and Sons, 1995.

[20] Koch, M, and Koch, J. Using component technology for group editors -The Iris Group Environment". In: *Proceedings of ECSCW'97*, OOGP,Workshop, 1997.

[21] MacLeod, I.S. Real-world effectiveness of Ergonomic methods. *Applied Ergonomics*, 34, 5 (September 2003), 465477.

[22] Millen, D.R. Rapid ethnography: time deepening strategies for HCI field research. In: *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*, NY, August 2000, 280-286.

[23] Pfautz, J., and Roth, E. Using cognitive engineering for systems design and evaluation: A visualization aid for stability and support operations. *International Journal of Industrial Ergonomics*, 36, 5 (May 2006), 389-407.

[24] Pinheiro, M.K., Lima, J.V., e Borges, M.R.S. A framework for awareness suport in groupware systems. *Computers in Industry*, 52, 1, 2003, 47-57.

[25] Pressman, R.S. *Software Engineering: a practitioner's approach*. 5th ed. McGraw-Hill, New York, 2000.

[26] Rasmussen, J. *Information processing and human-machine interaction: an approach to cognitive engineering*. Elsevier Science, New York, 1986.

[27] Singleton, W.T. *Man-machine systems*. Harmondsworth, Penguin, 1974.

[28] Sommerville, I. *Software Engineering*. 5th ed. Addison-Wesley, 1996.

[29] Souza, C.S. de, Leite, J.C., Prates, R.O., Barbosa, S.D.J. Projeto de Interfaces de Usuário: Perspectivas Cognitivas e Semióticas. *Anais da Jornada de Atualização em Informática, XIX Congresso da Sociedade Brasileira de Computação*, Rio de Janeiro, 1999. Disponível em http://www.dimap.ufrn.br/~jair/piu/JAI_Apostila.pdf. Acesso em Junho 2006.

[30] Suchman, L. *Plans and situated actions:the problem of human-machine communication*.Cambridge University Press, Cambridge, 1987.

[31] Wilson, J.R., Jackson, S., and Nichols, S. Cognitive Work Investigation and Design in Practice: The influence of social context and social work artefacts. In: Hollnagell, E. (Ed.), Handbook of Cognitive Task Design. Lawrence Erlbaum Associates, London, 2003, 83-98.

[32] Winograd, T., and F. Flores. Understanding Computers and Cognition. Ablex Publishing, Norwood, NJ, 1986

.

A Process for User Interaction Analysis in Collaborative Environments

Sandra de A. Siebra
Centro de Informática – UFPE, Caixa Postal 7851, Cidade Universitária, CEP 50732-970 Recife – PE – Brasil
(55-81) 21018430
sas@cin.ufpe.br

Ana Carolina Salgado
Centro de Informática – UFPE, Caixa Postal 7851, Cidade Universitária, CEP 50732-970 Recife – PE – Brasil
(55-81) 21018430
acs@cin.ufpe.br

Patrícia A. Tedesco
Centro de Informática – UFPE, Caixa Postal 7851, Cidade Universitária, CEP 50732-970 Recife – PE – Brasil
(55-81) 21018430
pcart@cin.ufpe.br

RESUMO

As interações que ocorrem em Ambientes de Aprendizagem Colaborativa (AAC) são um dos mais importantes aspectos visíveis da colaboração. A análise dessas interações pode dar suporte aos processos de reflexão e auto-avaliação dos estudantes, assim como às atividades dos professores. Neste cenário, esse artigo apresenta um processo para análise de interações que primeiro estrutura as interações, levando em conta o contexto onde elas estão ocorrendo. Posteriormente, ele armazena as interações em uma estrutura multidimensional e, finalmente, usa consultas OLAP para explorar e analisar as informações armazenadas sob diferentes perspectivas (dimensões), de acordo com as necessidades dos usuários. O principal objetivo desse processo é prover *feedback* imediato de qualidade para estudantes e professores. Esse artigo também apresenta uma implementação de parte desse processo e alguns resultados iniciais dos testes realizados com o mesmo. Os resultados iniciais se mostraram promissores.

ABSTRACT

Interactions taking place in Computer-Supported Collaborative Learning Environments (CSCLE) are one of the most important visible aspects of collaboration. The analysis of these interactions can provide support for students' reflection and self-regulation processes and for teachers' activities. In this light, this paper presents a process for interaction analysis that first structures the interactions, taking into account the context where the interactions are occurring. After that, it stores the interactions in a multidimensional structure and, finally, uses OLAP queries to explore and analyse the stored information from different perspectives (dimensions), according to the user needs. The main goal of this process is provide good quality just-in-time feedback for both students and teachers. This paper also presents an implementation of part of this process and some first results of the tests. The initial tests have indicated promising results.

Palavras-Chave

Análise de Interações, Aprendizagem Colaborativa, Contexto, CSCL

Keywords

Interaction Analysis, Collaborative Learning, Context, CSCL

1 INTRODUCTION

Collaboration is the co-construction of knowledge with the mutual engagement of participants [31]. Collaborative Learning is a strategy in which small teams, each with students of different levels of ability, use a variety of learning activities to improve their understanding of a subject. As Vygotsky [49] pointed out, in a collaborative scenario, students interchange their ideas for coordinating when they work for reaching common goals. When the learners work in groups they reflect upon their ideas (and those of their colleagues), explain their opinions, consider and discuss those of others, and as a result, they learn. Thus, each learner acquires individual knowledge from the collaborative interaction.

In general, CSCLE (Computer Supported Collaborative Learning Environments) can offer greater opportunities to share and solicit knowledge through the interaction. In most situations, externalization, articulation, argumentation, negotiation of multiple perspectives, are considered to be the main mechanisms that can promote collaborative learning [4][35][46][47]. In this light, interactions taking place in CSCLE are one of the most important visible aspect of collaboration and the analysis of these interactions could provide information to teachers (so that they are able to follow the evolution of students in the short, medium and long terms) and to students so that they can improve their knowledge and reflect about it.

However, the evaluation, organisation, storage and management of information related to the interactions that occur in CSCLE are complex tasks. Furthermore, in order to understand the meaning of the user interactions it is important to discover and register the context where each interaction

SBSC Simpósio Brasileiro de Sistemas Colaborativos
20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 19-28.

occurred [8]. Context is a collection of relevant conditions and surrounding influences that make a situation unique and comprehensive [7]. Thus, to fully understand many interactions, actions or events, it is necessary to access relevant contextual information [1][7][40]. For example, understanding why a student is finding it difficult to complete a task or to answer a question depends on what his/her knowledge level is, whether his/her workgroup is good or bad, what is the difficulty level of the task or question. In the interaction analysis the more details the system can provide about user's interactions, the better support it can give to teachers and students in the reflection process and to complete their activities based on historical informations.

Although context is a relevant aspect in the learning process, to the best of our knowledge, there are no CSCLE explicitly using the concept of context in their development or during the process of interaction analysis. Moreover, many CSCLEs do not store the interactions in a repository so that the stored interactions can be easily explored, can be viewed from different perspectives and could be presented selectively, according to users' needs. The current means of interaction storage (e.g. log files) can make difficult the interaction analytical analysis.

In this state of affairs, this paper discusses the benefit of interaction analysis, shows some related works and presents a process for user interaction analysis in CSCLE with some implementation issues. In order to do that, the remainder of this document is structured as follows: Section 2 discusses the benefits of interaction analysis. Section 3 presents some proposals for interaction analysis. Section 4 presents our process for interaction analysis. Section 5 shows some implementation issues and our first results and, finally, section 6 presents our conclusions and further works.

2 BENEFITS OF INTERACTION ANALYSIS

Despite the fact that recent information and communication technologies have favoured the appearance of several CSCLE, such environments still suffer from student's lack of motivation and high rates of evasion. Students need feedback and guidance from the system to learn effectively in groups [43][50]. Moreover, they need some tool that supports them in the reflection, self-regulation and self-evaluation process.

In this scenario, an extremely important role played by teachers is to guide and evaluate the students, as well as to mediate and encourage discussions in CSCLE. However, this task can be cumbersome when the number of participants and quantity of interactions increase. Furthermore, teachers generally do not have enough information to decide when it is the best moment to interact with students, or to evaluate their learning process [31]. Consequently, the need for tools that support the teachers (an aspect almost always neglected) in assessing and observing the students interaction and individual progress comes to light. Indeed, most CSCLE do not support teachers in their activities and necessities in the environment. For example, most CSCLE do not provide reports about the students' progress or about their performance within the environment or during chat sessions with the teacher. Such systems almost always offer support only for the student and such support is often quite generic or limited (for example, only some quantitative values). In other words even though the environments offer resources (e.g. communications tools and simple reports), these are often not enough for learners and teachers to work together in improving the learning process.

The support for teachers and students could be provided by using information that can be acquired through the evaluation and diagnosis of interactions that occur among students and/or students and teacher. Analysing the interactions will allow teachers, for example, to: (1) find out both the most difficult and most interesting subjects for a specific student or group of students; (2) follow the students' evolution, finding the quicker learners in each subject, so that they can be used as peer tutors for students having difficulties; (3) detect apathetic students who need to be motivated. Moreover, a mechanism to evaluate the interactions could help the users to reflect on, to discuss about a specific theme and to collaborate with other users in the environment.

The lack of a mechanism to evaluate the interactions could make it more difficult for users to focus their discussion on a specific topic or collaborating among themselves.

3 RELATED WORKS

Interaction Analysis has been acknowledged as a key issue in Collaborative Learning research [15], as a means for supporting student self-regulation as well as formative evaluation processes [33]. There are several proposals of tools for both approaches: [23][48] are examples of the first, whereas [3][32] exemplify the latter. Some authors state that these two approaches rely on the same basic functions [23][25], and that information obtained by teacher with an evaluation tool, for example, can be presented to students to achieve, promote self-regulation [2].

Usually, systems perform interaction analysis by applying methods from Artificial Intelligence that compare the results with an ideal case, and produce messages to guide students [13][45]. Alternatively, researchers also do an off-line analysis of the interaction in order to understand the collaborative processes [19][47].

Other related line of research is the work on Collaborative Learning Ontologies (e.g. [21]) that construct an ontology to represent CSCL sessions. The ontology will work as a vocabulary to describe the session and provide design patterns referred to during the instructional design process. In their work, Inaba et al. also present some instructional design support systems using the ontology as the TIA (Theory-based Interaction Analysis). The TIA support system will help human users to analyse complex interaction process in collaborative learning. It will be useful to interpret what type of collaborative learning is occurred in the learning session and to identify why a learning session is not effective.

In summary, up to now, the interaction analysis approaches present the following weaknesses:

- ✓ To generate feedback based in the interaction analysis, most of time, just to the student (excluding the teacher's needs).

- ✓ To only generate pre-defined reports and they don't allow cross informations to create new reports;
- ✓ To neglect the context where the interactions occur. The analysis should take into account the context where interaction emerges – the social, cultural and organisational factors that affect interaction, and on which the user will draw when making decisions about actions to take and in interpreting the CSCLE's and/or user's response. Consequently, identifying the contextual elements (e.g. user knowledge level, group abilities, exchanged message, related task) relevant to characterise, enrich and qualify the interaction is very important.
- ✓ To have inadequate mechanisms for interactions' persistence - group members may find it difficult to recall and justify their decisions when using interaction mechanisms with low or no persistence. Important information may be lost or need to be reproduced several times in order to achieve the desirable level of common knowledge [5]. Although many CSCLE [13][16][18][45] provide a way to store previously sent or received messages (e.g. by using sequential log files, normally organized in temporal order), what it is really needed is a common space (a group memory) to store the information in order to comfortably refer to it, add new contributions and analyse the stored information;

An interesting work that it is being done in the area of interaction analysis is the work of the project named JEIRP (Jointly Executed Integrated Research Project). They intend to define a Unified Framework on Interaction Analysis [19], considering an interaction as an action that affects or can affect the collaborative process. This framework will be based on defined concepts and relations, taking into account different application contexts (e.g. different kinds of technology based learning environments, simple and complex problem solving collaborative activities, existence or not of teachers, and educational level).

4 A PROCESS FOR USER INTERACTION ANALYSIS

In order to analyse user interactions so that we can support students' reflection process and teacher in his/her activities we have developed a four-stage process (see Figure 1). In our proposal, we have tried to overcome the weaknesses in the interaction analysis approaches described in the last section. Each stage (see Figure 1) of the process will be described in the following subsections. The subsection 4.1 presents the first step of the process (structuring interactions), which uses an argumentation model to classify and relate the interactions. Subsection 4.2 describes how we are mapping and using the concept of context. Subsection 4.3 presents our multidimensional model that gives rise to a group memory that contains all interactions occurred in the environment. This group memory will be called Learning Interaction Memory (LIM). Finally, subsection 4.4 shows the last step of the process where the interactions presents in the LIM are analysed and some feedback is generated to the students and teachers, based in this analysis.

4.1 Structuring Interactions

In Collaborative Learning, the ideas of Vygotsky [49] and his followers have brought up evidence that we can learn more when doing it in a group. Interacting with our peers gives us a forum to discuss our ideas, to take a stand on our views, to reflect about and to elaborate on them. Consequently, CSCLE tend to generate a large volume of interactions and the communication tools must be adapted in a way that facilitates and organizes the discussion, and supplies instruments to reduce the information overload on people [20].

A way to achieve some overload reduction is to structure the discussion and supplying simple and representative pieces of information about the messages' contents (as means to that helps the identification of message's relevance and context). Some authors [27][28][34] claim that classifying utterances:

- ✓ Increases the efficiency of online communications by reducing the cost of message production (saving typing);

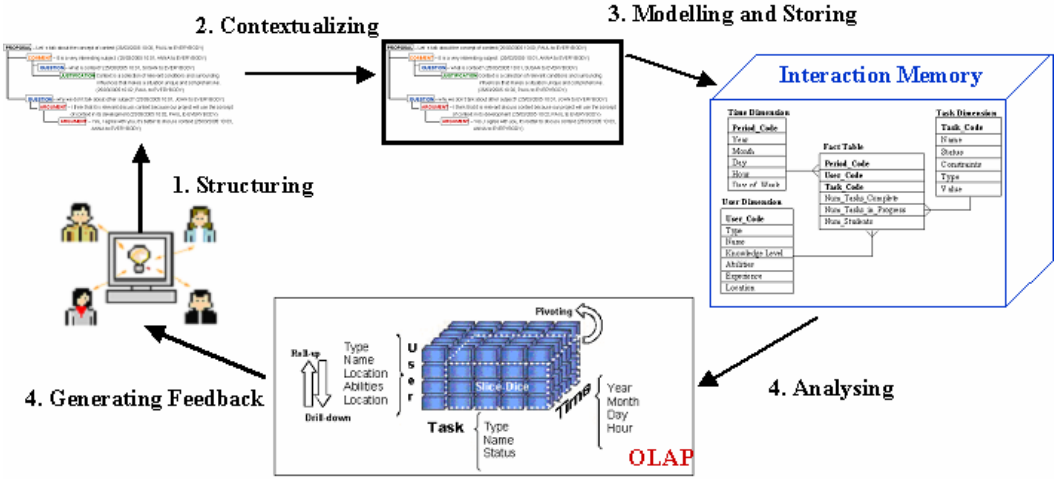


Figure 1. Stages of the Process for User Interaction Analysis

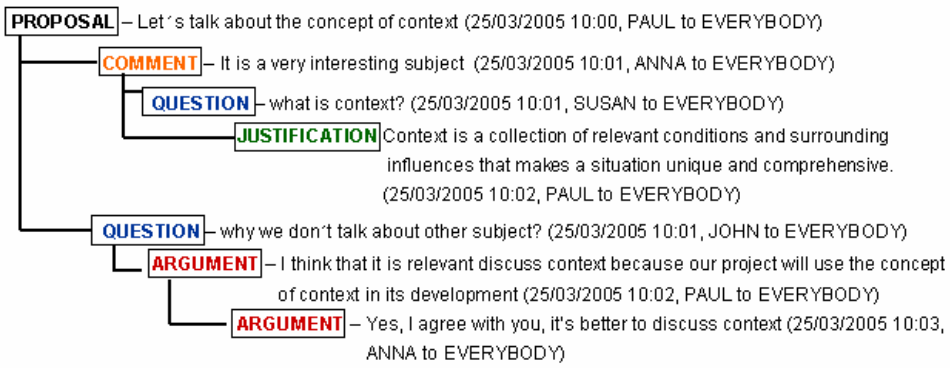


Figura 2. Example of a discussion graph

- ✓ Increases reflection on the part of interlocutors by requiring that they identify the intent of their communications and thereby;
- ✓ Increases the success of online communications by making intentions more explicit;
- ✓ Reduces the participants' information overload, since it supplies information that helps the identification of the discussion content and structure without read the messages.

Other benefits may arise from automated guidance enabled by tracking the dialogue. The choosing of a category requires an additional effort in the preparation of a message. In fact, the author's concepts, points of view and ideas must be well thought out in order to express her/himself in a way that separates her/his discourse into different messages, each one with its own category. Additionally, the structuring can help to improve the comprehension of problems, making it easier for participants to clearly organise their ideas, better understand others' points of view and arrive at a consensus more quickly.

In view of the above, structuring the interactions is the first step to achieve the interaction analysis because it can make explicit the intention of the messages and the relations among them. Thus, in order to structure the interactions, in this our process, we have proposed an Argumentation Model for collaborative discussions [41] based on the IBIS argumentation model [28].

Argumentation Models have been widely employed as a means to help people to capture and structure informal knowledge [28]. Moreover, they have often been applied as a mean of systematizing the communication patterns (whether synchronous or asynchronous) between members of a group. The use of an argumentation model implies in classifying all elements of discussions in pre-defined abstractions of the model (e.g. Argument, Explanation, etc) and connecting them via a set of pre-defined relations (e.g. provokes, generates, etc.). Upon sending a message, participants have to select the message category that reflects their intention more closely. In other words, users need to select one of a pre-defined set of appropriate utterance categories (e.g. question) that provide explicit information about the content of their messages. For example, "QUESTION – Do you know what an argumentation model is?". Consequently, the computational environment will be able to take advantage of the semantic knowledge of the categories and explicit relations among messages to organise and infer information.

Using an argumentation model, all the user interactions in the CSCLE's will be mapped to a discussion graph with a hierarchical structure (see Figure 2). Thus, the result of this stage is a complete discussion graph. Based in this discussion graph, CSCLE can generate reports and statistics about how the participants use the categories, which can help teachers to understand behavior and check compliance with tasks.

4.2 Contextualizing Interactions

In the learning process the more details the system can provide about the users' interactions, the more it can support their reflection and knowledge construction [1]. These details are the contextual information. Context is a collection of relevant conditions and surrounding influences that make a situation unique and comprehensive [7]. In conversation, context plays a fundamental role in disambiguating utterances. In situations where geographically separated individuals have to collaborate (especially asynchronously), technological support for understanding and storing the contextual information involved is very important. For instance, the choice of an example must be relevant in the student's socio-cultural environment. This identification of contextual information can help to clarify users' utterances, as well as to repair misinterpretations and provide adequate feedback.

In CSCL, we need to identify different types of contexts at different levels, trying to reach all the elements related to CSCLE and trying to answer the questions that summarise the definition of context [6]:

- ✓ Who? - Information about people
- ✓ When? - Information about Time and Historical Information
- ✓ Where? - Information about environment
- ✓ How? - Information about user's action plans
- ✓ What? - Information about users activities in progress
- ✓ Why? - Information about the reasons related to the user actions

These contexts have not the same granularity [10] and make difficult a simple representation of the contextual cues in our

process. Thus, based in the generic conceptual framework for analysing the use of context in groupware proposed by Rosa, Borges and Santoro [39], we have organised context in five different categories [42], described in the following. This organization will be useful to map out the information that it must be captured to qualify each interaction.

1) Information about scheduled tasks (Subject and Task Contexts) - in CSCLE it is important to know what the users are doing, what is the subject of the users are currently studying or about what they are discussing. So, this category includes:

   ✓ Subject Context – it consists of information about the subject that the user are currently studying or discussing about;

   ✓ Task Context - in CSCLE several tasks are possible (for example, to study a lesson, do exercises, do tests, discuss about a subject, share files, or draw something). The elements in this category are necessary to keep information about these tasks in the LIM in order to identify what an individual or group is or was doing.

2) Information about people and groups (Individual and Group Contexts) - The knowledge about the characteristics of individuals and the group as a whole is a resource that can be used by teachers to encourage interaction and collaboration [36]. This category includes:

✓ Group Context – it is important have in the LIM some knowledge about the group to understand the evolution of the individuals in CSCLE;

✓ Individual Context - the elements in this category help to characterise the user, as well as let other users better understand her/his doubts, difficulties and actions in the CSCLE. Some of the contextual information in this category can be obtained from the user's model (generally present in the CSCLE).

3) Information about the relationship between people and tasks (Interaction and Planning Context) - in CSCLE it is important to know who is doing what, i.e. what the task's execution plan and what is being discussed into the environment. Indeed the interaction analysis is important for discovering more about the student (e.g. her/his difficulties or doubts). This type of information is represented in two kinds of context:

   ✓ Interaction Context - it has information about the interactions occurred into the environment and about users' behavior when interacting;

   ✓ Planning Context – it consists of information about the course execution plan (generally present in the pedagogical model of CSCLE). The Planning Context could be implemented using the idea of proceduralised context and contextual graphs presented in [9][10]. The interest of contextual graphs is to do not limit the representation of a problem solving to the solution identified by the teacher, but to account for all the variants of the learners that lead also to the same solution. This is the difference between the official plan (made by the teacher) and the practices developed by students when they "contextualize" the plan in order to tailor the problem solving to the context at hand. In learning, an interesting side-effect of this approach is to identify clearly when a learner goes towards a dead-end way before the learner be in it;

4) Information about the environment where the interaction takes place (Session and Environment Context) - it consists of information that characterises the environment and the current session where the interaction takes place and influences task completion;

5) Information about tasks and activities already concluded (History Context) - the information in this category tries to characterise the interactions that have already occurred. Its goal is to provide background information about the experiences learned either from the same group or from similar tasks performed by other groups. In this category, all contextual information generated is stored for future retrieval. It is the repository of the "group memory" (including contextual elements). In this way a situation can be reconstructed with the context in which it occurred. Since the LIM provides historical context, students and/or teachers can access past incidents. This can also be used to share the latest news, seek advice and compare notes. Thus, it might be a source of reflection for both the teacher and the student.

Thus, each interaction structured in the discussion graph (that it was generated in the last stage) will be enriched with contextual information. Some contextual information will be captured implicitly by sensors and monitoring programs (e.g. a Perceptor Agent) included in the communication tools components of the CSCLE. Other contextual information will be captured explicitly (e.g., applying a questionnaire in the beginning of the interaction session, if the user is logged in the communication tool for the first time).

4.3 Modelling and Storing Interactions

After interactions are structured and contextualized, they can be stored so as to enable a later analysis. Without persistence, the structured and contextualized interactions are ephemeral and cannot be shared afterwards with people who were not involved at the time it occurred [4]. With persistence, the interactions will constitute an Interaction Memory (IM) mechanism, which make past information about the interactions readily and selectively available when required. The IM is the result of a process of accumulating data generated by group members during discussions in synchronous or asynchronous tools after the stages of structuring and contextualization.

This kind of historical information will be extremely useful to the analysis stage because it allows following the changes of social roles, beliefs, conflicts, misconceptions, and views of the task within the group. Indeed, this dialogue history is viewed as an important resource in collaborative dialogue since it provides a common reference to previous activity (unlike most spoken dialogues) that may encourage reflection and more effective collaboration [12]. This kind of information would help teachers to track students' evolution process. Besides, the analysis of the interaction memory would enable users to reuse historical information to solve future problems, reminding

participants of previous ideas (encouraging elaboration on them) and possibly serving as an agenda for further work.

For the purpose of facilitating the execution of analytical analysis, this IM can be modelled in a multidimensional structure [26]. With multidimensional modelling the IM's information can be viewed from different perspectives (e.g. information can be easily crossed or filtered) and could be presented selectively, according to users' needs (i.e. depending on their context, users could access different information).

4.4 Analysing Interactions in a Multidimensional Way for Generating Feedback

Multidimensional Analysis is a process that involves organising and summarising data in a multiple number of dimensions. In the case of CSCLEs, the interactions data can be organised through multidimensional models that represent a conceptualization of the interaction data that is closer to the way in which the students perceive the discussion.

OLAP [22][26] is the technology that enables users or applications to efficiently access data in a multidimensional way. OLAP allows a system to operate on-line queries on the multidimensional data, enabling the user to create sophisticated inquiries. Data can be analysed (summarised, consolidated, consulted and/or processed) along multiple levels of abstraction and from different perspectives (dimensions), according to the user needs.

OLAP queries can be used as much by the users (directly) as by the CSCLE to support students and teachers and generate reports giving details of how the discussion is progressing and the different types of learner contributions. In this way, it will be possible to answer questions such as: which kind of knowledge has been shared within the environment? Which members are participating more actively? Which are the most frequent problems that have been found during the learning processes via the environment? Which topics are being more difficult to students? Which students are not motivated? Which students have already faced the problem Y, considering the context X?

5 IMPLEMENTATION ISSUES

For the purpose of test some ideas, we have implemented a prototype to put in practice the described process. To simplify, we have used an Intelligent Chat for Collaborative Discussion (SmartChat) in the place of the CSCL environment, considering that a chat environment is a basic component of communication in any CSCL environment. Whereas the prototype's architecture can be seen in the Figure 3.

5.1 The Architecture

Figure 3 shows our tool's agent-based architecture. It consists of an *Agent Society* composed by two agents (*Perceptor* and *Modeller Agents*) and four components: *Loader Component, Report Generator, Domain Knowledge Base* and *the Contextual User Model*. The Agent Society is responsible by structures, contextualizes and stores structured information in a *XML Log* file, during the session of discussion. A session of discussion starts when the chat room is opened and the students log in and finished when all students log out.

In the end of the session, the Agent Society group the interactions related to that session and store them in a *Learning Interaction Memory (LIM)* implemented using a Data Warehouse. The use of Data Warehouse allows *Analytical Queries* be executed and that historical information about the discussions in the CSCL environment be kept. The response of the analytical queries helps the environment to generate feedback for both, students and teachers. In the following paragraphs, we will describe this agent-based architecture in details.

5.1.1 SmartChat

The chat engine was implemented using RMI (Remote Method Invocation) [29]. It was developed in Java [29] and uses a hybrid knowledge base of rules and objects accessed by the JEOPS inference motor [17]. The domain ontology was implemented in XML [37] and it covers the subject "Object-Oriented" [44]. SmartChat's user interface is quite simple and it is only available in Portuguese, in this first version of the prototype

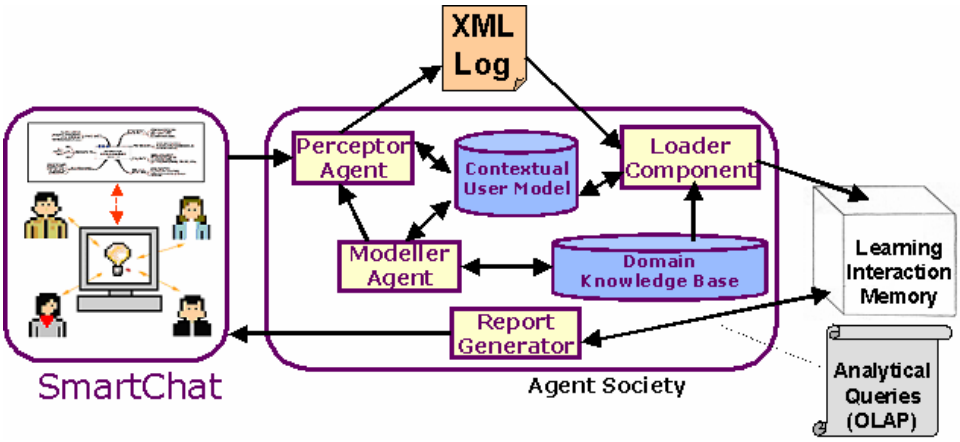


Figure 3. Prototype's Architecture

5.1.2 The Agent Society

It consists of an Agent Society composed by two agents (*Perceptor* and *Modeller Agents*) and four components: *Loader Component, Report Generator, Domain Knowledge Base* and *the Contextual User Model*.

The *Perceptor Agent* is responsible for monitoring all user interactions occurred in the learning environment and, also, structures and contextualizes these interactions. In order to do this, it has the following main functions:

1) Monitor the user interactions;

2) Structure the interactions using an argumentation model. To categorize his/her interaction, the user should choose one abstraction (e.g. Question, Argument, Proposal, Justification or Maintenance) that qualifies his/her interaction. However, since the user sometimes does not choose the adequate argumentation model abstraction to categorize her/his message, the Perceptor Agent tries to infer the intention underlying each utterance. In order to do so, it uses a parser and a predefined vocabulary of word categories (for example, the abstraction question can have words such as: "what", "why", "when", "how"). Whereas the interactions were categorized, the Perceptor Agent infers the relation among them using the argumentation model (e.g. there is a relation between the abstractions Justification and Question called responds_to).

3) Capture the initial individual context of the user connect to the chat - if the user is accessing the environment, for the first time, the Perceptor Agent shows to the user a questionnaire to get information about him/her (individual context). This information is stored in the Contextual User Model, where all the information about the users that have already used the chat environment is stored. If is not the first time that the user is accessing the environment, the Perceptor Agent, based in the login and password of the user, load the information store in the Contextual User Model about him/her. The Contextual User Model is implemented in XML.

4) Take into account the context of the interactions (according to the context categories described in the section 4.2) before put the interaction in the XML log file. To contextualized the utterance in this point, it will be used the Task, Group, Individual, Planning and Interaction contexts. The Task, Group and Planning Contexts are captured in the Domain Knowledge Base. The Individual Context is in the Contextual User Model. The Interaction Context is captured by this agent.

5) Generate the XML log of the interactions taking place in the environment with the contextualized and related (using the argumentation model) interactions;

The Perceptor Agent was implemented as a thread [29] and constantly monitors the messages sent among users. With each message sent, the Perceptor executes a parser that processes the utterance and searches in the Domain Ontology for concepts related to the subject under discussion. The Domain Ontology is in the Domain Knowledge Base and it configures the vocabulary related to the subject to be discussed.

In the process of data warehousing, this agent is responsible by the steps of Extract and Clean [26], considering that it prepares the informations that the Loader Component will use to load the data warehouse.

The *Modeller Agent* is responsible for supporting students' reflection as well as teachers' decisions and also for classifies the users according to their participation level in the discussions. Thus, it has the following main functions:

1) Classify the users' participation;

2) Infer a stereotype to the user according to the Contextual User Model and the user's participation in the discussion. In order to classify the user, the Modeller Agent was implemented as a thread and "wakes up" each time a group of ten messages arrives at the chat server. This agent uses production rules (located in the Contextual User Model) and the inference motor called JEOPS [17] to classify the user as one of the stereotypes: challenger, agreer, remiss, tutor, contributor, questioner, and unnatentive.

3) Discover the best strategy to adopt in order to promote reflection and also generate reports to the users (student and teachers), based on the analysis of the contextualized interactions, using the information in the XML Log, in the Learning Interaction Memory (through analytical queries) and in the Contextual User Model.

4) Update the Contextual User Model of the users logged-in according to their behavior in the chat session.

5) Decide at what moment to interfere in the conversation - The identification of the user's stereotype allows the Modeller Agent to interfere in the discussion to perform one of the three following actions: (a) send support messages to the users (e.g. "Well done! You've got excellent explanations!" or "You are so laconic, wouldn't you like to participate more?"), according to the stereotype s/he fits in; (b) suggest references related to the subject being discussed (e.g.,. "If you have doubts about this subject, then you should consult the X book"); and (c) name another chat user that may collaborate with the user having difficulties (e.g., "Look, maybe Joe can help you with your doubts. Why don't you talk to him?"). In motivating the student and striving to stimulate the collaboration between users, we try to make the interactions in SmartChat more effective and focused in the subject being discussed.

The *Loader Component* is responsible for uploading the Learning Interaction Memory – LIM (implemented using Data Warehouse). In the data warehousing process, this component is responsible by the Load step. For this purpose, in each end of chat sessions, it takes, processes and stores in the LIM the information obtained from XML Log file, the Contextual User Model and the Domain Knowledge Base.

Finally, the *Report Generator Component* is responsible for sending information to the CSCL environment. It is also responsible for issuing different statistical reports for teachers and students during and at the end of the discussion session. The teacher report presents general statistics about all the

students, indicating the most discussed topic, the most and the less active students, and the topic that caused more conflicts. The student report presents her/his individual performance in relation to the group's. It is also possible have access to the interaction history to see the information storage there.

The *Contextual User Model* has many components: the User Model (an model based in stereotypes [38]), the users individual context (store in a XML file), the production rules to classify the users participation (according to the User Model) and the production rules to discover the best strategy to adopt to promote reflection to each user stereotype. Users will be model through the following stereotypes: challenger, agreer, remiss, tutor, contributor, questioner, and unnatentive. Students will be classified in one of these categories by the Modeller Agent. The stereotypes are based on the idea that the users belong to homogeneous groups and describe their performance during the chat session.

The *Domain Knowledge Base* is composed by the Domain Ontology (see Figure 4). Initially, the Domain Ontology employed deals with the subject Object-Oriented [44]. It was implemented using XML. It acknowledges word variations used in the discussion (plurals, abbreviations, marks and synonyms) and each subject in the ontology has registered its difficulty level, pre-requirements and some indicated reference (e.g. book or link).

```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
- <ontologia>
  - <orientacao>
    - <objet>
      - <variacao>
          <obj />
          <object />
        </variacao>
        <preRequisito>nenhum</preRequisito>
        <grauDificuldade>médio</grauDificuldade>
      + <constr>
      + <destr>
      + <oper>
      - <atrib>
        - <vis>
            <public />
            <priv />
            <proteg />
          </vis>
        </atrib>
      </objet>
    + <class>
    + <mecan>
    </orientacao>
  </ontologia>
```

Figure 4 - Example of the Domain Ontology

5.2 The Prototype

In the first version of the prototype, we have implemented stages 1 to 3 of our four-stage process (see Figure 1), in other words, we have structured, contextualized, modelled and stored the interactions in the LIM (implemented using data warehouse). However, we have not yet explored the LIM using OLAP. This prototype is being developed in an iterative and incremental way [27] to keep up with the process stages.

Some feedback is already given to the teacher and students based on the information in the Contextual User Model and the XML Log file, using the JEOPS inference motor [17] and a hybrid knowledge base of rules. Using the rules, the agent society can identify collaboration opportunities (indicating someone to help you if you are in troubles), recommend references (books or links registered in the ontology to each subject in it) or give some motivation message.

5.3 First Results

An important step in the experimentation of the process was to perform a test with the prototype to check: (1) if the users would be adequately classified; (2) the correctness of the discussion graph; (3) the correctness and the level of acceptance of the feedback produced; and (4) the environment usability.

To perform the test, ten people were chosen to participate of a discussion session using SmartChat. The requirements to select a participant were, basically, that he/she has some knowledge about Object-Oriented (domain of our ontology).

During two hours of chat, discussing the chosen themes, the users could test all the functionalities of the environment and observe its weak and strong points. First, it was observed that the system fulfilled the expectancy. The analysis of the interactions took place correctly, all users were classified according to the profile presented during the session, and received appropriate feedback from the Modeller Agent. Also the discussion graph was generated correctly and the contextual information was correctly captured (we have checked the files of the Contextual User Model and XML log).

However, the users have suggested that the feedback should be improved to offer more information and statistics about the chat and to help them in their self-evaluation. This will be possible when the four stage of the process will be implemented, in the next version of the prototype.

Moreover, it was possible to confirm that the muldimensional model was correctly filled in.

6 CONCLUSIONS AND FUTURE WORKS

Research [14][46][48] has shown that students learn effectively in groups, where one encourages the other by asking, answering, reasoning, explaining, justifying their opinions and reflecting about what is being discussed. Thus, the interaction becomes the key element to understand the process of building knowledge and the role of each student in this process. In this scenario, the interaction analysis can produce resources for the system to support individual and group needs as well as to better assess their behavior, attitudes, knowledge level, and difficulties.

In this paper, we have presented a process for Interaction Analysis. This process uses an argumentation model to categorise and structure the group interactions as means to make explicit the intention of the messages and the relations among them. In the second stage of the process, contextual

information is added to each structured interaction in order to provide the correct cues to give the right interpretation to it. Subsequently, the contextualized interactions are modelled and stored in a multidimensional structure (called Learning Interaction Memory) for the purpose of facilitating the execution of analytical analysis. Finally, in the last stage, the interaction memory is explored (by users or CSCLE) using OLAP queries [22] to support students and teachers and generate many kinds of feedback reports.

We also have presented a context-based analytical tool prototype, which it has allowed to evaluate in the practice, with users, the ideas and constructed artifacts (e.g. discussion graph and multidimensional model).

Our future research will concentrate on implementing the OLAP queries to explore the Learning Interaction Memory and provide good quality just-in-time feedback for both students and teachers. We also intend to include a feedback area in the prototype's interface, in order to inform the students about their performance in relation to the group.

7 ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

8 REFERENCES

[1] Araujo, R.; Brézillon, P. Reinforcing shared context to improve collaborative work. *Revue d'Intelligence Artificielle. Special Issue on "Applying Context Management"*. S. Schultz, T. Roth-Berghofer & P. Brézillon (Eds.), 19, 3, 2005, 537-556.

[2] Avouris, N. M.; Dimitracopoulou, A.; Komis, V. On analysis of collaborative problem solving: An object-oriented approach. *Journal of Human Behavior*, v. 19, n. 2, 2003, 147-167.

[3] Avouris, N. M.; Dimitracopoulou, A.; Komis,V.; Fidas, C. OCAF: An object-oriented model of analysis of collaborative problem solving. *Proceedings CSCL 2002*, Boulder, Colorado. Hillsdale, NJ: Lawrence Erlbaum 2002.

[4] Baker, M. A Model for Negotiation in Teaching Learning Dialogues. *International Journal of Artificial Intelligence in Education*, v. 5, n. 2, 1994, 199-254.

[5] Borges, M. R. S.; Pino, J.A. Requirements for Shared Memory in CSCW applications. *Proceedings of 10th Annual Workshop On Information Technologies and Systems (WITS'2000),* Brisbane, Australia, 2000, 211-216.

[6] Bazire, M.; Brézillon, P. Understanding context before to use it. Modeling and Using Context (CONTEXT-05), A. Dey, B.Kokinov, D.Leake, R.Turner (Eds.), *Springer Verlag, LNAI 3554*, 2005, 29-40.

[7] Brézillon P. Context in problem solving: A survey. *The Knowledge Engineering Review*, v. 14, n.1, 1999, 1-34.

[8] Brézillon P. Making context explicit in communicating objects. C. Kintzig, G. Poulain, G. Privat and P-N. Favennec (Eds.), *Communicating Objects*, Hermes Science Editions, Lavoisier, 2002.

[9] Brézillon, P. Context dynamic and explanation in contextual graphs. Modeling and Using Context (CONTEXT-03), P. Blackburn, C. Ghidini, R.M. Turner and F. Giunchiglia (Eds.). *Springer Verlag, LNAI 2680*, 2003, 94-106.

[10] Brézillon, P. Representation of procedures and practices in contextual graphs. *The Knowledge Engineering Review,* 18, 2, 2003, 147-174.

[11] Brézillon, P.; Borges, M.R.S.; Pino, J.A.; Pomerol, J.C. Context-based Awareness in Group Work. *Proceedings of the 17th International Flairs Conference*, Miami Beach, Fl, USA, 2004.

[12] Collins, A; Brown, J.S. The computer as a tool for learning through reflection. H. Mandl and A. Lesgold (Eds). *Learning Issues for Intelligent Tutoring Systems,* Springer-Verlag, New York, 1988, 1-18.

[13] Constantino-González, M.; Suthers, D. Coaching Collaboration by Comparing Solutions and Tracking Participation. Eds. P. Dillenbourg, A. Eurelings & K. Hakkarainen. *Proceedings EuroCSCL 2001*, Maastricht, The Netherlands, 2001, 173-180.

[14] Dillenbourg P. What do you mean by collaborative learning? Ed. P. Dillenbourg. *Collaborative-learning: Cognitive and Computational Approaches*, Oxford: Elsevier, 1999, 1-19.

[15] Dillenbourg P.; Self J.A. Designing human-computer collaborative learning. Ed. C.E. O'Malley. *Computer Supported Collaborative Learning*, Springer-Verlag, Hamburg, 1995, 245-264.

[16] Eleuterio M., Barthès JP, Bortolozzi F. Mediating collective discussions using an intelligent argumentation-based framework. *Proceedings of Computer Supported Cooperative Learning* (CSCL 2002), Boulder, Colorado, USA, 2002.

[17] Figueira Filho, C.; Ramalho G. JEOPS – Java Embedded Object Production System. Monard, M. C; Sichman, J. S (Eds). Advances in Artificial Intelligence, International Joint conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI, (IBERAMIA-SBIA 2000), *Proceedings. Lecture Notes in Computer Science 1952*. Springer, 2000, 53-62.

[18] Fuks H.; Gerosa, M.A.; Lucena, C.J.P. Using the AulaNet Learning Environment to Implement Collaborative Learning via Internet. In: *Innovations 2003 - World Innovations in Engineering Education and Research*, iNEER, USA, 2003.

[19] Fuks, H.; Laufer, C.; Choren, R.; Blois, M. Communication, coordination and cooperation in distance education. *Proceedings of the Americas Conference on Information Systems*, Milwaukee, USA, 1999, 130-132.

[20] Fussel, S. R.; Kraut, R. E.; Lerch, F. J.; Scherlis, W. L.; Mcnally, M. M.; Cadizz, J. J. Coordination, Overload and Team Performance: Effects of Team Communication Strategies. *Proceedings of CSCW 1998*, 1998, 275-284.

[21] Inaba, A., Tamura, T., Ohkubo, R., Ikeda, M., Mizoguchi,

R., & Toyoda, J. 'Design and Analysis of Learners' Interaction based on Collaborative Learning Ontology *Proceedings of Euro-CSCL2001*, 2001, 308-315.

[22] Inmon, W.H. *Building the data warehouse* (2nd ed.). New York: John Wiley & Sons, 1996.

[23] Interaction & Collaboration Analysis. Supporting Teachers & Students. Self-regulation Unified Framework on Interaction Analysis. Available in: http://www.rhodes.aegean.gr/LTEE/KALEIDOSCOPE-ICALTS/Publications/D2%20Unified%20Framework%20ICALTS%20Kal%20www.pdf

[24] Jermann, P. Task and Interaction Regulation in Controlling a Traffic Simulation. Ed. G. Stahl. *Proceedings of CSCL2002*, Boulder, Colorado. Hillsdale, NJ: Lawrence Erlbaum, 2002, 301-302.

[25] Jermann, P.; Soller, A.; Mühlenbrock, M. From Mirroring to Guiding: a Review of the State of the Art Technology for Supporting Collaborative Learning. *Proceedings of Euro-CSCL'2001*, Maastrich, Holland, 2001.

[26] Kimball, R.; Reeves, L.; Ross, M.; Thomthwaite, W. *The data warehouse lifecycle toolkit: tools and techniques for designing, developing and deploying data warehouses*. New York: John Wiley & Sons, 1998.

[27] Kruchten, P. *The Rational Unified Process*. 2nd Edition. Addison Wesley Longman Inc., 2000.

[28] Kunz, W.; Rittel, H. Issues as Elements of Information Systems. *Technical Report, Paper No. 131. Universität Stuttgart*, Institut für Grundlagen der Plannung, (1970).

[29] The Source for Java Technology – Available in: <http://java.sun.com>. (Accessed in July 2006).

[30] Levin, J.; Moore, J. Dialogue games: meta-communication structures for natural language interaction. *Cognitive Science*, v. 1, 1977, 395-420.

[31] Levy, P. *Les technologies de l'intelligence. L'avenir de la pensée à l'ère informatique*. La découverte, 1990.

[32] Lipponen, L. Exploring foundations for computer-supported collaborative learning. Ed. G. Stahl. Computer Support for Collaborative Learning: Foundations for a CSCL community. *Proceedings of the Computer-supported Collaborative Learning 2002*. Hillsdale, NJ: Erlbaum, 2002, 72-81.

[33] Martinez, A.; Dimitriadis, Y.; Gómez, E.; Rubia, B.; De la Fuente, P. Combining qualitative and social network analysis for the study of classroom social interactions. *Computers and Education*, v. 41, n. 4, 2003, 353-368.

[34] Martinez, A.; Marcos, J.; Garrachon, I.; de la Fuente, P.; Dimitriadis, Y. Towards a data model for the evaluation of participatory aspects of collaborative learning. *Proceedings of the Conference on Computer Support for Collaborative Learning, Workshop on Designing Computational Models of Collaborative Learning Interaction,* Boulder, Colorado. Hillsdale, NJ: Lawrence Erlbaum, 2002, 45-51.

[35] McManus, M.; Aiken, R. Monitoring computer-based collaborative problem solving. *Journal of AI in Education,* v. 6, n. 4, 1995, 307-336.

[36] Pinheiro, M. K.; Lima, J. V.; Borges, M. R. S. A framework for awareness support in groupware systems, *Computers in Industry*, Netherlands, v. 52, n. 1, 2003, 47-57.

[37] Ray, E. T. *Learning XML*. O'Reilly and Associates, 2001.

[38] Rich, E. Stereotypes and user modeling. Eds. A. Kobsa & W. Wahlster, *User Models in Dialog Systems*, Springer-Verlag, Berlin, Heidelberg, 1989, 35–51.

[39] Rosa, M. G. P.; Borges, M. R. S.; Santoro, F. M. A Conceptual Framework for Analyzing the Use of Context in Groupware. *Proceedings of International Workshop on Groupware*, Autrans, France, *Lecture Notes in Computer Science*, Berlin, Germany, v. 2806, 2003, 300-313.

[40] Santoro, F. M.; Brézillon, P. The role of the shared context in group storytelling. *Computers And Informatics, Special Issue on "Groupware, Context and Design"*. M.R.S. Borges, H. Fuks, S. Lukosh and J.A. Pino (Eds.) (To apprear), 2006.

[41] Siebra, S. A.; Christ, C.; Queiroz, A. E.; Tedesco, P.; Barros, F. SmartChat – An Intelligent Environment for Collaborative Discussions. *Proceedings of ITS 2004,* Maceió, Al, 2004.

[42] Siebra, S. A; Salgado, A.C.; Tedesco, P. A. Structuring Participants' Interactions in Collaborative Learning Environments. *Proceedings of WebMedia/LA-Web 2004 - WCSCW*, Ribeirão Preto, S, Brasil, 2004, 181- 187.

[43] Siebra, S. A; Salgado, A.C.; Tedesco, P. A.; Brézillon, P. A Context-based Analytical Environment for CSCLE. Internal Report, CIN – UFPE, Recife – PE. Available in: http://www.cin.ufpe.br/~sas/reports ,2004.

[44] Sintes, A. *Aprenda Programação Orientada a Objetos em 21 dias*, Makron, 1a. edição, 2002.

[45] Soller, A.; Linton, F.; Goodman, B.; Lesgold, A. Toward Intelligent Analysis and Support of Collaborative Learning Interaction. *Proceedings of the Ninth International Conference on Artificial Intelligence in Education,* LeMans, France, 1999, 75-82.

[46] Soller, A.; Wiebe, J. ; Lesgold, A. A Machine Learning Approach to Assessing Knowledge Sharing During Collaborative Learning Activities. *Proceedings of Computer Support for Collaborative Learning*, Boulder, CO, 2002, 128-137.

[47] Veerman, A. L. *Computer-Supported Collaborative learning through argumentation*. University of Utrecht, published PhD. Thesis, 2000.

[48] Vieira, A. C. H. *Classificando automaticamente diálogos colaborativos on-line com o OXEnTCHÊ-Chat.* Dissertação de Mestrado. Cin-UFPE, Recife – PE, 2004.

[49] Vygostki, L. *Mind and society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press, 1978.

[50] Zumbach, J.; Hillers, A.; Reimann, P. Supporting Distributed Problem-Based Learning: The use of Feedback mechanisms in online learning. Eds. T. Roberts. *Online Collaborative Learning: Theory and Practice*. Hershey, PA: Idea, 2003.

Avaliando os Elementos de Contexto para as Atividades de Coordenação em Pré-reuniões

Márcio G. P. Rosa
Faculdade Ruy Barbosa
Salvador/BA
marciorosa@frb.br

José Maria N. David
Faculdade Ruy Barbosa
Salvador/BA
josemaria@frb.br

Flávia Maria Santoro
Departamento de Informática Aplicada,UNIRIO
flavia.santoro@uniriotec.br

Marcos R.S. Borges
Núcleo de Computação Eletrônica, UFRJ
mborges@nce.ufrj.br

ABSTRACT
Designing groupware in order to support pre-meetings activities, relevant information need to be supplied to the participants of a group aiming to increase awareness. Some of them are related to the context information about participants and their activities. During this meeting cycle phase, awareness elements design could be supported by a conceptual framework in order to present to the participants important information related with a specific context. The goal of this paper is to describe the results obtained through the application of a conceptual framework throughout awareness elements design of web-based pre-meeting application. Such results show the importance of a framework deployment in contextual elements design, as well as the need to provide extensions so that other asynchronous activities could be supported as, for example, the coordination activities.

RESUMO
No projeto de groupware para apoiar as atividades de pré-reuniões, informações relevantes precisam ser fornecidas aos participantes de um grupo para potencializar a percepção. Algumas delas estão relacionadas à informação sobre os contextos nos quais se inserem estes participantes e as atividades das quais eles participam. Nesta fase do ciclo de reuniões, o projeto dos elementos de percepção poderia ser apoiado por um framework conceitual no intuito de oferecer aos participantes informações referentes aos diferentes contextos envolvidos. O objetivo deste artigo é descrever os resultados obtidos através da aplicação de um framework conceitual no projeto de elementos de percepção em uma aplicação de suporte a pré-reuniões na web. Tais resultados sinalizam a importância da utilização de um framework no projeto de elementos de contexto, bem como a necessidade de extensões do mesmo para que outras atividades assíncronas possam ser apoiadas como, por exemplo, as atividades de coordenação.

Keywords
Context, Pre-meeting, Awareness, Coordination

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 29-37.

Palavras-chave
Contexto, Pré-reunião, Percepção, Coordenação

1. INTRODUÇÃO
Reuniões são geralmente realizadas quando decisões precisam ser tomadas ou informações necessitam ser distribuídas. Pesquisas têm sido desenvolvidas com o objetivo de propor soluções para melhorar não apenas as interações entre os participantes, mas a coordenação em diferentes fases do ciclo de reuniões. Algumas delas estão relacionadas à condução da fase de pré-reuniões para reduzir o tempo e o custo, e para aumentar a produtividade [10][18].

No projeto dessas aplicações, a colaboração poderia ser potencializada se fosse disponibilizado um conjunto mais abrangente de mecanismos de percepção para apoiar a complexidade das interações.

Dourish e Bellotti [16] definem percepção como sendo a compreensão das atividades realizadas por outras pessoas, capaz de fornecer aos participantes do grupo informações importantes relacionadas ao contexto de suas atividades. Sem conhecimento do contexto, participantes do grupo não sabem o que os outros estão fazendo, podendo isto muitas vezes acarretar retrabalho e inconsistências, ou então impedir que comunicações espontâneas possam ser iniciadas. Através da contextualização das atividades, contribuições relevantes podem ser fornecidas para o grupo e avaliações podem ser realizadas com informações sobre a evolução do grupo.

A hipótese considerada é que ao se fornecer elementos de contexto mais adequados para apoiar aplicações de pré-reuniões, a percepção poderia ser potencializada. Assim, é proposta a utilização de um framework conceitual como um guia para apoiar o projeto de elementos contextuais em aplicações de pré-reuniões.

Em uma primeira etapa elementos de percepção, projetados sem o auxílio de um framework conceitual, foram identificados. Em seguida, estudos de caso foram conduzidos com o intuito de identificar as divergências e novos elementos para serem adicionados a este framework. Posteriormente, tais elementos foram implementados incluindo aqueles que poderiam oferecer informações de contexto para os coordenadores. Para avaliar a hipótese, outro estudo de caso foi conduzido tendo como foco, as atividades de coordenação e os elementos necessários para fornecer informações de contexto. Adicionalmente, propôs-se que este apoio poderia ser estendido para o projeto de outras aplicações do ciclo de reuniões.

Este artigo está organizado da seguinte forma: a Seção 2 discute alguns requisitos de percepção e a implementação de aplicações de pré-reuniões. Na Seção 3 o relacionamento entre elementos de contexto e percepção é descrito. A Seção 4 analisa os elementos de contexto em uma aplicação específica. Na Seção 5, os experimentos de avaliação são descritos. Por fim, a Seção 6 conclui o trabalho e sugere alguns trabalhos a serem realizados.

2. GROUPWARE PARA APOIO À PRÉ-REUNIÕES

Muitas reuniões não são bem sucedidas devido à falta de planejamento e de preparação. A literatura de groupware propõe sistemas de apoio à pré-reuniões com o objetivo de aumentar a produtividade das reuniões [3] [10] [18]. Algumas questões, consideradas básicas, quando respondidas nesta fase, podem minimizar a ocorrência de erros, otimizar os recursos alocados e tornar a próxima etapa mais produtiva. Dentre estas questões, podem ser citadas: (i) quais os participantes da reunião; (ii) quais os papéis de cada participante?; (iii) quais as técnicas e métodos que serão utilizados nas discussões?; (iv) qual será a agenda e o tempo disponível?; (v) quais os resultados esperados? [11].

Benefícios adicionais também podem ser obtidos, tais como: documentação para a reunião poderia ser previamente acessada; e atividades básicas poderiam ser detectadas e realizadas a fim de conduzir reuniões posteriores, por exemplo, convidar participantes imprescindíveis para as discussões. Como resultado, reuniões poderiam ser mais objetivas e o seu tempo poderia ser diminuído. Alguns recursos são oferecidos aos participantes da pré-reunião com o objetivo de apoiar a troca de idéias e aumentar a participação.

Sistemas como o SISCO [3][10] e o PRIME [18] foram implementados para apoiar pré-reuniões assíncronas como objetivo de preparar participantes para a discussão dos itens da próxima fase.

Em ambos os sistemas, contribuições disponibilizadas durante a pré-reunião (questões, posições argumentos e comentários) são armazenadas na memória de grupo. Assim, o apoio à percepção no passado é fornecido posteriormente com a apresentação de eventos no espaço de trabalho compartilhado, a partir do momento em que o usuário é autenticado no sistema. Para cada contribuição ele é informado se ela é nova, ou então se ela é nova e nunca foi lida.

Ao acessar o PRIME o usuário visualiza todas as pré-reuniões abertas das quais ele participa. Associadas a ela é possível encontrar: a sua descrição, a quantidade de contribuições, e os itens a serem discutidos.

O sistema de apoio à pré-reunião COPSE-Web [14] foi desenvolvido de acordo com uma infra-estrutura oferecida para apoiar o projeto de aplicações de groupware na web. Além disso, são disponibilizados pela infra-estrutura elementos que podem ser instanciados para apoiar a comunicação, a coordenação, a percepção e a memória de grupo.

Dentre as funcionalidades implementadas na aplicação de pré-reunião do COPSE-Web pode-se mencionar (i) definição da pré-reunião – nome e descrição, objetivos, prazos e itens para ser discutidos posteriormente. Informações do grupo também são fornecidas, tais como, participantes e seus papéis; (ii) acompanhamento dos itens de discussão – as contribuições dos participantes, e; (iii) acompanhamento da preparação dos itens – os participantes responsáveis pela preparação da atividade, seu percentual de conclusão, prioridade e prazo para a conclusão.

Adicionalmente, um conjunto de elementos contextuais está disponível para os participantes de tal forma que os objetivos da reunião possam ser mais claros e, como resultado, os participantes podem construir uma lista de itens para serem discutidos na fase de reunião. Posteriormente, discussões irrelevantes ou redundantes podem ser evitadas.

Entretanto, durante interações mediadas por sistemas de pré-reuniões, membros de um grupo necessitam acessar um contexto compartilhado mais rico. Em interações assíncronas, este aspecto se torna claro considerando que é necessário definir as informações que devem ser armazenadas. Posteriormente tais informações precisam estar disponíveis para outros participantes no espaço de trabalho compartilhado. Neste momento, provavelmente eles terão diferentes conhecimentos e informações contextuais. Sem recuperar as informações relacionadas às interações ocorridas no passado, participantes da pré-reunião, mais especificamente os coordenadores, podem ter dificuldade para perceber o contexto, alterar suas posições e comportamentos. Por exemplo: a partir do momento em que um participante saiu da aplicação, novos itens de discussão surgiram, outros participantes foram convidados e novos documentos foram disponibilizados. Ao retornar à pré-reunião, agora com conhecimentos diferentes, ele necessita, prontamente, perceber esses elementos contextuais para aproximar diferentes contextos.

A importância da persistência das interações para recuperar e analisar elementos contextuais relevantes em ambientes de apoio à aprendizagem colaborativa é discutida de uma forma mais abrangente em [24]. Ao armazenar as interações que ocorrem nestes ambientes, participantes de um grupo podem visualizá-las em diferentes perspectivas, e, como resultado, contextualizá-las em processos de aprendizagem.

3. MODELO DE CONTEXTO PARA GROUPWARE

Na literatura relacionada a groupware, são observadas diversas referências ao termo contexto, normalmente associadas ao conjunto de informações necessárias para que os membros do grupo possam se tornar conhecedores do ambiente onde estão atuando. As áreas de interação humano-computador [8] e de inteligência artificial [7], bem como aquela relacionada às aplicações context-aware [15] já elaboraram suas definições para contexto.

Contexto e percepção devem ser considerados sempre em conjunto no domínio de groupware [9]. Contexto é a representação do conhecimento que envolve uma situação, incluindo as informações e eventos necessários para tal. Percepção é um dos mecanismos que o groupware utiliza para prover informações de contexto para o grupo. Cada membro pode perceber a mesma informação de contexto de maneira diferente, pois o ato de perceber está associado com a cognição do indivíduo.

Quando os membros de um grupo estão trabalhando colaborativamente, contexto se torna especialmente relevante, , a

ausência de recursos para acesso a suas informações pode gerar um trabalho truncado, sem coesão, não representando as idéias do grupo como um todo, mas somente um conjunto de idéias soltas, com pouca ou nenhuma ligação entre elas, gerando inconsistências e contradições.

Alguns autores definiram contexto em atividades colaborativas [5][20][21]. Rittenbruch [20], por exemplo, considera que o contexto é uma descrição do conhecimento sobre as circunstâncias nas quais uma atividade e seus eventos ocorrem. Naturalmente, este contexto deve ser representado de acordo com o modelo do usuário, e explicado a fim de que seja acessível e compreensível por cada participante.

Contexto pode ser considerado como uma construção dinâmica composta de cinco dimensões: (1) tempo, (2) casos de uso, (3) interações sociais, (4) objetivos internos, e (5) influências locais [17]. Embora os elementos contextuais em algumas situações sejam estáveis, compreensíveis e previsíveis, isto não é sempre verdade. Casos com aparentemente o mesmo contexto podem ser diferentes. Membros do grupo não podem interagir efetivamente se eles não são capazes de compreender as informações contextuais que estão ao redor e influenciam a interação [1][2][12].

A fim de reduzir a dificuldade para selecionar informações contextuais relevantes em uma atividade apoiada por groupware, modelos de representação de contexto são sugeridos [25]. Propõe-se a utilização de um framework conceitual para identificar e classificar os elementos contextuais mais freqüentes em groupware [21]. Um framework conceitual é representado por quadros conceituais, cujo objetivo é fornecer diretrizes para pesquisa e desenvolvimento [23].

Informações contextuais são organizadas em cinco categorias principais: (1) informação sobre pessoas e grupos; (2) informação sobre as atividades programadas; (3) informação sobre o relacionamento entre pessoas e atividades; (4) informação sobre o ambiente onde a interação acontece, e; (5) informação sobre atividades já concluídas. Estas categorias foram derivadas do Modelo Denver [22] e, em cada uma buscou-se identificar os aspectos de interação mais relevantes que influenciam o desempenho das atividades do grupo.

A primeira categoria se relaciona às informações sobre os indivíduos e os grupos aos quais eles pertencem. O conhecimento das características dos indivíduos e dos grupos é importante para o bom desenvolvimento das atividades em equipe, pois estimula a interação e a comunicação informal entre os membros, facilitando a cooperação [19]. Posteriormente esta categoria é dividida em dois tipos de contexto. O primeiro é o contexto individual que fornece informação sobre o indivíduo que é membro do grupo. O outro é o contexto do grupo que fornece informações sobre as características do grupo. Este dado é similar ao contexto individual, mas agora relacionado ao grupo.

A segunda categoria é a informação sobre as tarefas a serem realizadas. Independentemente da forma como a interação ocorre, membros do grupo precisam compreender as características das tarefas. Contexto da tarefa é o nome dado a este contexto.

A terceira categoria diz respeito ao relacionamento entre membros do grupo e as tarefas agendadas. Seu objetivo é relacionar as ações de cada membro durante as interações com as tarefas e suas atividades correspondentes. No escopo deste framework, a interação que se desenvolve entre os membros do grupo se inicia com um plano de execução e termina quando a tarefa encerra, passando pela seqüência de ações necessárias para desenvolver o plano. Em algumas situações a interação pode ser interrompida antes que a tarefa seja concluída. A razão para esta conclusão prematura é também parte do contexto e é relevante para a compreensão do que justificou a interrupção. Para essa categoria, são propostos dois tipos de contexto, o primeiro deles é o contexto da interação, composto de informações que representam o curso das ações ocorridas durante a realização das tarefas. Essas informações emergem no momento em que as pessoas começam a interagir, realizando ações para alcançar seus objetivos. Já o segundo tipo é o contexto do planejamento, composto de informações que representam o plano de ação da equipe.

A quarta categoria reúne informações sobre o ambiente. Representa os aspectos relacionados ao local onde a interação acontece. Ela abrange tanto os aspectos organizacionais quanto o ambiente tecnológico. Isto é, toda a informação fora do projeto, mas dentro da organização, que pode afetar a maneira na qual as tarefas são realizadas. O ambiente fornece algumas indicações adicionais aos membros do grupo sobre como a interação ocorrerá.

A última categoria reúne todas as informações sobre as tarefas concluídas. Seu objetivo é fornecer informações de background sobre as experiências aprendidas pelo mesmo grupo ou tarefas similares realizadas por outros grupos. Ela deve incluir toda a informação contextual sobre os projetos prévios. Denomina-se este conjunto de informação de "contexto histórico".

A Tabela 1 resume os sete tipos de contexto e apresenta seus objetivos bem como os elementos contextuais associados que podem influenciar as atividades do grupo.

O framework conceitual é uma classificação genérica de elementos contextuais. Ele não trata todas as peculiaridades de um domínio particular nem se aplica a um tipo especifico de groupware. Por ser genérico pode ser um ponto de partida para uma classificação de elementos contextuais para domínios particulares, como ferramentas de apoio a pré-reuniões, nos quais novos elementos podem ser identificados e considerados como relevantes.

4. ELEMENTOS DE CONTEXTO DE COORDENAÇÃO

Buscando avaliar contexto em sistemas de pré-reunião, propôs-se como primeiro passo uma análise baseada no *framework* conceitual apresentado [21]. A aplicação escolhida foi a Pré-reunião COPSE-Web. Tentou-se identificar os elementos de contexto disponíveis neste groupware de acordo com o *framework* (Tabela 1). A seguir, foram realizados dois estudos de caso para avaliar a relevância destas informações, bem como a necessidade de acrescentar novos elementos.

Tabela 1 – Framework conceitual para a análise de contexto em aplicações de groupware [21]

Tipo de Informação	Contextos Associados	Objetivo	Exemplos de informações contextuais	
Pessoas	Indivíduo	Identificar os indivíduos através da apresentação de suas características e dados pessoais.	• Nome • Habilidades • Interesses • Formação • Experiência	• Organização • Local de Trabalho • Horário de trabalho • Dados pessoais • Página pessoal
	Grupo	Identificar as equipes através da apresentação das suas características.	• Nome • Componentes • Papéis • Habilidades • Interesses	• Experiência • Estrutura organizacional • Sede geográfica • Horário de trabalho
Tarefas a serem realizadas	Tarefa	Identificar as tarefas através da apresentação das suas características.	• Nome • Descrição • Objetivo • Prazo • Pré-requisitos	• Homem/hora • Ações a serem realizadas • Restrições • Tecnologia envolvida
Relações entre pessoas e tarefas	Interação (Síncrono)	Representar, de forma detalhada, o curso das ações ocorridas durante a realização da tarefa.	• Grupo envolvido • Mensagens trocadas pelo grupo • Noção de presença • Gestos executados pelos indivíduos	• Ações realizadas • Objetivo de cada ação • Justificativa de cada ação • Autor de cada ação
	Interação (Assíncrono)	Representar, através de uma visão ampla, o curso das ações ocorridas durante a realização da tarefa.	• Grupo envolvido • Ações realizadas • Autor de cada ação • Objetivo de cada ação	• Justificativa de cada ação • Versões dos artefatos • Data de realização de cada ação.
	Planejamento	Representar o plano de execução da tarefa a ser realizado	• Papéis na interação • Regras do plano • Metas • Responsabilidades	• Estratégias • Procedimentos de coordenação • Plano de trabalho
Ambiente	Ambiente	Representar o ambiente onde as interações ocorrem. São características do meio que influenciam a execução das tarefas.	• Padrões de qualidade • Regras do ambiente • Procedimentos padronizados • Estratégias padronizadas • Prazos institucionais	• Estrutura organizacional • Decisões políticas • Restrições financeiras • Plataforma de hardware e software
Tarefas já realizadas	Histórico	Oferecer subsídios aos componentes do grupo no entendimento do processo de realização de uma tarefa já concluída.	• Nome da tarefa • Objetivo da tarefa • Plano de trabalho • Ações realizadas • Autor de cada ação • Objetivo de cada ação • Justificativa de cada ação	• Data de realização de cada ação • Versões de artefatos • Informações dos outros contextos que tenham sido utilizadas de forma relevante na execução da tarefa.

Na Tabela 2 são apresentados os elementos de contexto identificados na versão de Pré-reunião COPSE-Web analisada, chamada versão A. 'OK' significa que a aplicação disponibiliza a informação de contexto, já o 'X ' que o elemento está ausente no groupware. Nesta versão, os mecanismos que fornecem estas informações foram desenvolvidos sem nenhum apoio de um framework conceitual. Todos os elementos foram baseados em estudos e análises em ferramentas já desenvolvidas. Na primeira avaliação observou-se uma carência de informação em vários dos tipos de contexto propostos. Em contrapartida, foram percebidos alguns elementos de contexto que não estavam descritos. Os elementos em questão estavam relacionados com apoio à atividade de coordenação, como a quantidade de contribuições e a participação de cada participante na pré-reunião, e a identificação de conflitos indesejáveis [13].

A partir desta primeira análise percebeu-se que as atividades de coordenação deveriam ser melhoradas. Uma descrição mais detalhada desta análise encontra-se em [13]. Este foi o ponto de partida para a definição de um conjunto de informações de contexto a ser acrescentado ao framework apresentado, ou seja, o contexto de coordenação. Esta avaliação pôde ser obtida através de estudos de casos.

Para aumentar a efetividade da coordenação, alguns mecanismos foram melhorados principalmente em relação à interface como, por exemplo, o participômetro e o contribuitômetro [6]. Tais mecanismos foram disponibilizados ao coordenador no sentido de identificar os percentuais de participação e de contribuição. Através destes percentuais, o coordenador pode identificar aqueles participantes que não estão preparados, ou então, estão desmotivados para a atividade. Quanto mais cedo o coordenador puder identificar tais índices, atitudes poderão ser tomadas no sentido de corrigir eventuais falhas.

Outros mecanismos foram implementados, tais como o "impactômetro" e o "conflitômetro" [6]. O primeiro mecanismo, juntamente com o "contribuitômetro", busca qualificar as contribuições pelo impacto causado por elas. Na aplicação de pré-reunião este impacto está associado à quantidade de mensagens geradas por uma contribuição. Já o segundo busca auxiliar o coordenador na identificação de conflitos indesejáveis como por exemplo, aqueles que podem polarizar uma atividade de discussão, ou desviar o seu foco [6].

Além desses, outros mecanismos foram projetados para a geração de agrupamentos de dados na pré-reunião de acordo com as necessidades do coordenador. Tais mecanismos podem ser criados durante a pré-reunião e disponibilizados também a todos os participantes, se necessário. Dentre eles cabe citar os gráficos para a visualização dos percentuais das categorias de mensagens (questões, posições e argumentos) por participante. Desta forma, diferentes coordenadores podem direcionar as atenções para percentuais – e as suas evoluções – ao longo da discussão. Além de reduzir a sobrecarga de informações no espaço de trabalho, tais agrupamentos têm como objetivo direcionar a atenção dos participantes para informações relevantes.

Na nova versão da aplicação de Pré-reunião COPSE-Web, denominada versão B, também foram disponibilizados outros elementos de contexto, além dos mecanismos de percepção citados anteriormente. Dentre eles podemos citar aqueles relacionados ao contexto individual e ao contexto da interação, conforme apresentado na Tabela 2. Outros elementos não foram implementados nesta versão, como por exemplo, aqueles relacionados ao contexto do ambiente.

O segundo passo na avaliação de elementos de contexto em sistemas de pré-reunião se deu através de um terceiro estudo de caso, já com a versão B do sistema de Pré-reunião COPSE-Web.

Neste momento a atenção estava voltada para as informações que permitissem apoiar as atividades de coordenação. Considerou-se também que diferentes coordenadores devem possuir mecanismos para acompanhar o andamento da pré-reunião, e intervir se necessário em diferentes momentos. Para tanto, procurou-se identificar se os novos elementos de contexto auxiliaram tais atividades

Durante interações assíncronas, nas quais existe mais de um coordenador, a necessidade de contextualização de uma maneira rápida e eficiente torna-se imperativa. Situações nas quais ocorrem perda e conflito de contexto [4] apontam para necessidades de contextualizações específicas para as aplicações de pré-reunião. Uma primeira situação na qual este fato pode acontecer é quando um coordenador se ausenta por um período de tempo e, no momento do seu retorno necessita entender, por exemplo, o motivo pelo qual um participante deixou de interagir. Outra situação na qual a necessidade de contextualizar as atividades do coordenador se faz necessária é quando dois coordenadores não entendem um contexto comum no qual eles estão atuando, e, como resultado, têm as atenções dirigidas para aspectos diferentes da pré-reunião. Por exemplo, um coordenador está acompanhando um eventual conflito entre dois participantes da pré-reunião. A partir de um determinado instante, este coordenador necessita interromper as suas atividades. Em seguida, um segundo coordenador dá continuidade às atividades de coordenação, porém não identifica prontamente tal conflito. Este fato se deve a inexistência de mecanismos para revelar possíveis conflitos indesejáveis, bem como a formas pelas quais eles foram resolvidos previamente.

Sistemas como o SISCO [3][10] oferecem apoio à coordenação através de mecanismos como o "participômetro" [6]. Tal mecanismo foi projetado para oferecer informações sobre o percentual de participações de cada membro do grupo. A partir dessas informações deve ser possível definir diretrizes para promover a discussão fornecendo condições para que o percentual de participação aumente, provavelmente através da motivação dos membros do grupo. Entretanto, quando alguns coordenadores estiverem ausentes por um período, ao retornarem eles precisam obter informações sobre as decisões tomadas previamente. Por exemplo, por que alguns itens da discussão estão com uma baixa participação, e quais as decisões já tomadas para aumentar este percentual?

Além dessas questões outras precisam ser consideradas, tais como: por que uma discussão não evoluiu? Por que outros participantes foram convidados a participarem?

5. AVALIANDO CONTEXTO EM SISTEMAS DE PRÉ-REUNIÃO

Para a condução da pesquisa, dois estudos de casos, anteriormente referenciados, foram inicialmente realizados. Em ambos a aplicação de pré-reunião foi avaliada considerando os mecanismos de percepção projetados anteriormente e os elementos de contexto inerentes a eles. Procurou-se também avaliar o *framework* conceitual de contexto para uma aplicação de pré-reunião.

Foram utilizadas fontes de dados quantitativas e qualitativas durante a coleta de dados, são elas: questionários, observações diretas e entrevistas após os estudos de caso.

No primeiro estudo de caso o grupo era composto de sete estudantes de graduação. Todos eles foram treinados em conceitos de groupware bem como em aplicações de suporte a reuniões. Ao longo de duas semanas, os participantes discutiram como melhorar infra-estrutura de laboratório de informática utilizado ao longo do curso. Durante esse período, foram identificados ajustes necessários no experimento (tempo previsto, tema de discussão, e tempo de treinamento) como também questões do questionário.

Tabela 2 – Elementos Contextuais identificados na ferramenta Pré-reunião COPSE-Web (adaptado de [21])

Contextos	Exemplos de elementos contextuais	COPSE-Web Pre-meeting (versão A)	COPSE-Web Pre-meeting (versão B)	Exemplos de elementos contextuais	COPSE-Web Pre-meeting (versão A)	COPSE-Web Pre-meeting (versão B)
Individual (Síncrono e Assíncrono)	Nome	**Ok**	**Ok**	Organização	X	X
	Habilidades	X	**Ok**	Local de trabalho	X	X
	Interesses	X	**Ok**	Horário de trabalho	X	X
	Formação	X	**Ok**	Dados pessoais	**Ok**	**Ok**
	Experiência	X	**Ok**	Página pessoal	X	X
Grupo (Síncrono e Assíncrono)	Nome	**Ok**	**Ok**	Experiência	X	**Ok**
	Componentes	**Ok**	**Ok**	Estrutura organizacional	X	X
	Papéis	**Ok**	**Ok**	Sede geográfica	X	X
	Habilidades	X	**Ok**	Horário de trabalho	X	X
	Interesses	X	**Ok**			
Tarefa (Síncrono e Assíncrono)	Nome	**Ok**	**Ok**	Homem/hora necessário	X	X
	Descrição	**Ok**	**Ok**	Ações a serem realizadas	**Ok**	**Ok**
	Objetivo	**Ok**	**Ok**	Restrições	X	**Ok**
	Prazo	**Ok**	**Ok**	Tecnologia envolvida	**Ok**	**Ok**
	Pré-requisitos	X	X			
Interação (Síncrono)	Grupo envolvido	X	**Ok**	Ações realizadas	X	**Ok**
	Noção de presença	**Ok**	**Ok**	Autor de cada ação	X	**Ok**
	Mensagens trocadas pelo grupo	**Ok**	**Ok**	Objetivo de cada ação	X	**Ok**
	Gestos executados pelos indivíduos	X	X	Justificativa de cada ação	X	**Ok**
Interação (Assíncrono)	Grupo envolvido	**Ok**	**Ok**	Justificativa de cada ação	X	X
	Ações realizadas	**Ok**	**Ok**	Versão de artefatos	X	X
	Autor de cada ação	**Ok**	**Ok**	Data de realização de cada ação	**Ok**	**Ok**
	Objetivo de cada ação	**Ok**	**Ok**			
Planejamento (Síncrono e Assíncrono)	Papéis na tarefa	**Ok**	**Ok**	Estratégias	X	**Ok**
	Regras no plano	X	X	Procedimentos de coordenação	X	**Ok**
	Metas	X	**Ok**	Plano de execução	**Ok**	**Ok**
	Responsabilida-des	X	**Ok**			
Ambiente (Síncrono e Assíncrono)	Padrões de qualidade	X	X	Prazos institucionais	X	**Ok**
	Regras do ambiente	X	X	Estrutura organizacional	X	X
	Procedimentos padronizados	X	X	Decisões políticas	X	X
	Estratégias padronizadas	X	X	Restrições financeiras	X	X
Histórico (Síncrono e Assíncrono)	Nome da tarefa	**Ok**	**Ok**	Autor de cada ação	**Ok**	**Ok**
	Descrição da tarefa	**Ok**	**Ok**	Objetivo de cada ação	**Ok**	**Ok**
	Objetivo da tarefa	**Ok**	**Ok**	Justificativa de cada ação	X	**Ok**
	Plano de execução	**Ok**	**Ok**	Data de realização de cada ação	**Ok**	**Ok**
	Ações realizadas	**Ok**	**Ok**	Informações dos outros contextos que tenham sido utilizadas de forma relevante na execução da tarefa	**Ok**	**Ok**

O segundo estudo de caso teve como objetivo avaliar as mesmas questões do primeiro, agora com os ajustes no experimento já realizado. Ou seja, o interesse era confrontar resultados obtidos previamente com o uso do *framework* e os elementos de contexto identificados na pré-reunião. Doze participantes interagiram durante aproximadamente duas semanas. O mesmo item proposto no primeiro estudo de caso foi novamente discutido. Alguns dos participantes começaram a discussão enquanto outros não estavam interagindo no período.No momento em que o segundo grupo foi convidado a participar, aproximadamente 50% do total dos itens de discussão já tinham sido postados., o segundo grupo de estudantes foi convidado a participar na discussão. Este fato permitiu avaliar a eficácia de alguns mecanismos de percepção assíncronos e as informações de contexto associadas a eles. De uma forma geral, estes resultados evidenciaram uma necessidade de revisão nas informações de contexto do grupo, bem como nos mecanismos de comunicação (Contexto de Interação). Além disso, ficou claro em todas as fontes de coleta de dados que as atividades de coordenação necessitavam de um apoio mais intenso principalmente se considerarmos a necessidade de diferentes coordenadores atuando em momentos diferentes. Tal fato pôde ser observado quando um coordenador relatou a dificuldade para comparar o percentual de participação e de contribuição, a partir do momento em que aumenta a quantidade de membros do grupo. Foi sugerido que tais informações fossem ordenadas de acordo com a necessidade do(s) coordenador(es), por exemplo. Os resultados destes experimentos encontram-se resumidos na Tabela 3 e discutidos com mais detalhes em [13].

A partir desses resultados novos mecanismos de percepção foram implementados considerando-se sobretudo o *framework* de elementos de contexto discutido previamente.

Tabela 3 – Resultados dos Estudos de Casos

Em percentual: (1) Discordo fortemente, (2) Discordo, (3) Não deseja opinar, (4) Concordo, (5) Concordo Fortemente.
Valor médio (μ) e Desvio padrão (δ).

Questões do Estudo de Caso	Estudo de Caso	(1)	(2)	(3)	(4)	(5)	μ	δ
Sobre a percepção das atividades (contexto da interação):								
(1) Os participantes tiveram dificuldade para identificar os papéis definidos.	1°	-	28.6	-	71.4	-	3.4	**0.90**
	2°	16.7	58.3	8.3	16.7	-	2.3	**0.92**
	3°	-	-	-	-	-	-	-
(2) Os mecanismos de percepção disponíveis permitiram identificar quais são os demais participantes da pré-reunião e em que ponto, exatamente, cada um estava trabalhando.	1°	28.6	14.3	-	42.9	14.3	3.0	**1.51**
	2°	-	25.0	-	58.3	16.7	3.7	**1.03**
	3°	-	16.7	-	50.0	33.3	4.0	**1.00**
(3) Em algum momento, não foi possível identificar uma determinada contribuição devido ao excesso de informação no espaço de trabalho.	1°	14.3	28.6	-	28.6	28.6	3.3	**1.49**
	2°	-	-	-	-	-	-	**-**
	3°	-	-	-	-	-	-	-
(4) Foi possível identificar, precisamente, onde cada participante estava trabalhando.	1°	-	-	-	-	-	-	**-**
	2°	8.3	25.0	8.3	58.3	-	3.2	**1.07**
	3°	-	33.3	-	33.4	33.3	3.7	**1.25**
(5) Foi possível identificar as contribuições de cada participante	1°	-	-	-	-	-	-	**-**
	2°	-	16.7	8.3	58.3	16.7	3.8	**0.92**
	3°				66.7	33.3	4.3	**0.47**
(6) Foi possível identificar as informações relacionadas à pré-reunião, tais como: objetivo, prazos e coordenador(es).	1°	-	-	-	-	-	-	**-**
	2°	-	66.7	8.3	16.7	8.3	2.7	**1.03**
	3°	-	16.7	16.6	66.7	-	3.5	**0.76**
Sobre as atividades de coordenação (contexto de coordenação):								
(7) Os mecanismos de percepção disponíveis para as atividades do coordenador/facilitador foram adequados para conduzir a pré-reunião.	1°	-	-	-	100.0	-	4.0	**-**
	2°	-	-	-	100.0	-	4.0	**-**
	3°	-	-	-	-	-	-	-
(8) Os mecanismos de percepção disponíveis para os membros do grupo permitiram uma participação efetiva.	1°	50.0	50.0	-	-	-	1.5	**0.5**
	2°	50.0	50.0	-	-	-	1.5	**0.5**
	3°	-	-	-	-	-	-	-
Sobre a comunicação (contexto da interação):								
(9) Foi possível estabelecer um canal de comunicação entre os participantes.	1°	14.3	28.6	-	28.6	28.6	3.3	**1.49**
	2°	-	-	25.0	58.3	16.7	3.8	**0.92**
	3°	-	-	-	50.0	50.0	4.5	**0.50**
(10) Os mecanismos de comunicação foram suficientes para estimulara cooperação e para gerar oportunidades de interação.	1°	-	71.4	-	28.6	-	2.6	**0,90**
	2°	-	-	-	-	-	-	**-**
	3°	-	-	-	-	-	-	-
(11) A categorização de mensagens utilizada na ferramenta contribuiu para flexibilizar a discussão	1°	-	28.6	-	57.1	14.2	3.6	**1.05**
	2°	-	8.3	8.3	50.0	33.3	4.1	**0.86**
	3°	-	-	-	33.3	66.7	4.7	**0.47**
(12) A utilização de um espaço de trabalho compartilhado facilitou o desenvolvimento das atividades de pré-reunião	1°	-	85.7	-	-	14.2	2.4	**1.05**
	2°	-	8.3	16.7	50.0	25.0	3.9	**0.86**
	3°	-	16.7	-	33.3	50.0	4.2	**1.07**

Em seguida, um terceiro estudo de caso foi realizado tendo como objetivo avaliar as mesmas questões apresentadas anteriormente; porém, com uma nova versão da aplicação de pré-reunião. Neste estudo um grupo composto por quatro alunos do programa de iniciação científica interagiu no sentido de avaliar e propor mudanças em relação ao programa do qual eles faziam parte. Além desses alunos dois coordenadores apoiaram as discussões, em períodos alternados. Tal fato teve como objetivo avaliar os elementos de contexto fornecidos aos coordenadores para apoiarem atividades assíncronas.

Ao término deste período todos os participantes responderam ao questionário proposto. Adicionalmente, os coordenadores responderam a um questionário específico no intuito revelar aspectos relacionados ao contexto de coordenação. Entrevistas e observações diretas também foram utilizadas neste experimento como fonte de coleta de dados.

5.1 Discussões

Considerando os dados coletados principalmente através dos questionários e resumidos na Tabela 3, as questões (2) e (4) evidenciam que os elementos de contexto implementados permitiram melhorar a apresentação das informações no espaço de trabalho compartilhado.

Particularmente a questão (2) nos sugere que as informações relacionadas ao contexto do indivíduo contribuíram de alguma forma para a realização da pré-reunião. Além disso, os resultados obtidos nas questões (5) e (6), em conjunto com as observações e as entrevistas, apontam para uma melhoria nas informações sobre o contexto da interação. Em ambas as questões observamos uma média (μ) relativamente alta e um desvio padrão (δ) baixo.

Chamaram a atenção também os valores obtidos na questão (11). Este resultado nos mostra que houve uma modificação em relação aos resultados anteriores. Tal fato pode estar relacionado não

apenas às alterações na apresentação das informações no espaço de trabalho, mas sobretudo ao treinamento realizado previamente. Esta observação foi confirmada pelas entrevistas informais ao término do terceiro experimento. Embora os métodos de coleta de dados utilizados sejam baseados em respostas subjetivas, a maioria dos participantes informou que os novos mecanismos de visualização do espaço de trabalho compartilhado também poderiam reduzir sobrecarga de informação. Entretanto, tal fato não pôde ser confirmado com os resultados da questão (12) ao apresentar média e desvio padrão relativamente altos. Esta informação nos leva a supor que alguns participantes tiveram dificuldades para interagir através do espaço de trabalho, ou então não identificaram precisamente o objetivo da questão. Tal fato pôde ser confirmado através das entrevistas quando alguns participantes relataram a dificuldade de visualização da discussão a partir do momento em que a quantidade de mensagens aumenta.

Além disso, um percentual alto de mensagens com conteúdos similares e pouco objetivas foi relatado por um coordenador. Provavelmente, isto se relaciona à dificuldade de visualização de mensagens e, consequentemente, a uma divergência de contextos principalmente entre aqueles que se ausentaram da pré-reunião por um período, e precisa ser confirmado em estudos posteriores.

Além do contexto da interação, as perguntas (5) e (11) nos levaram a supor que a visualização das contribuições e a categorização de mensagens utilizada foram suficientes para esta aplicação. Porém, a pergunta (2) sugere que são necessárias mais informações relacionadas ao contexto do indivíduo que contribuiu. Este fato é associado com o desvio-padrão relativamente alto obtido. Provavelmente, devem ser projetados mecanismos adicionais como formas alternativas de visualizar os relacionamentos entre as contribuições no espaço de trabalho compartilhado. A forma hierárquica de apresentação das informações não é a mais adequada para a visualização destes relacionamentos. Na realidade, esta sugestão foi confirmada pelas respostas e entrevistas. A maioria dos participantes informou que os novos mecanismos de visualização do espaço de trabalho compartilhado também poderiam reduzir sobrecarga de informação.

Em relação às atividades de coordenação, ambos os coordenadores informaram através de questionários que os mecanismos de apoio à percepção tais como, os relatórios de participação e de contribuição, bem como o mecanismo para informar os impactos das contribuições auxiliaram na condução das atividades.

Esta evidência também foi confirmada através das respostas a outras questões nos questionários e nas entrevistas. Nos questionários eles informaram que foi possível identificar prontamente os percentuais de contribuições e de participações de cada membro do grupo. Nas entrevistas, entretanto, foi informado que à medida que a quantidade de participantes aumenta, torna-se difícil relacionar tais contribuições e participações, bem como identificar os eventuais conflitos. Assim sendo foi sugerida uma ordenação dos percentuais para auxiliar as atividades do coordenador.

Também, em relação aos resultados anteriores, foi possível identificar que a forma pela qual as mensagens foram apresentadas no espaço de trabalho auxiliou nas atividades de coordenação. Nos estudos anteriores um percentual significativo de participantes relatou que na maioria das vezes se perdia o contexto no qual a mensagem foi inserida. A forma – hierárquica – de apresentação das mensagens no espaço de trabalho dificultava a identificação das novas mensagens principalmente quando a quantidade de mensagens aumentava. Para solucionar este fato a inserção de mensagens foi implementada em um outro espaço de trabalho. Ao mesmo tempo a hierarquia de mensagens foi mantida aberta indicando o contexto de inserção das mesmas.

6. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo discute aspectos do projeto de mecanismos de percepção em sistema de pré-reunião. A hipótese considerada é a de que a cooperação pode ser potencializada se elementos de contexto forem fornecidos de um modo sistemático. Assim, uma solução que considera um *framework* conceitual de contexto é discutida para prover diretrizes para projetos de groupware. Porém, o *framework* considerado não enfoca em particularidades de aplicações específicas. Assim sendo, identificou-se a necessidade de avaliar a aplicação deste *framework* para as aplicações de pré-reuniões. Inicialmente, os elementos de contexto mais importantes foram identificados, e percebeu-se através de experimento que havia uma necessidade de implementar novos elementos de contexto não contemplados na aplicação, bem como novos mecanismos de coordenação. Estes últimos poderiam apoiar as atividades de diferentes coordenadores que, muitas vezes, necessitam interagir assincronamente através de um mesmo espaço de trabalho. Como resultado eles precisam de informações capazes de aproximar os diferentes contextos percebidos por cada um.

Os resultados obtidos através de experimentos apontam para uma direção na qual elementos de contexto ainda precisam ser implementados, enquanto outros necessitam de melhorias principalmente em relação à interface. Dentre os elementos não implementados, cabe ressaltar aqueles relacionados ao contexto de ambiente. Certamente eles exercem uma influência sobre os outros elementos, e, portanto, poderiam enriquecer as informações oferecidas aos coordenadores. Percebe-se também uma necessidade de realizar outros experimentos considerando-se sobretudo um período maior de interação e outras situações de uso em diferentes grupos e organizações.

Além disso, é fundamental que sejam feitas outras avaliações com as aplicações das fases de reunião e pós-reunião, no intuito de revelar prováveis elementos de contexto ainda não identificados e implementados.

7. REFERÊNCIAS

[1] Alarcon, R. A., Guerrero, L. A., Ochoa, S. F., Pino, J.A. Context in Collaborative Mobile Scenarios. In Fifth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT´05), Workshop on Context and Groupware. CEUR Proceedings, Vol. 133, Paris, France, July 2005.

[2] Agostini, A., De Michelis, G., Grasso, M. A., Prinz, W., Syri, A. Contexts, Work Processes, and Workspaces, Computer Supported Cooperative Work: The Journal of Collaborative Computing, Kluwer Academic Publishers, Vol.5, No.2-3, 1996, 223-250.

[3] Bellassai, G., Borges, M. R. S., Fuller, D. A., Pino, J. A., Salgado, A. C. SISCO: A tool to improve meetings productivity. In Proc. of the I CYTED-RITOS International Workshop on Groupware, Lisbon, Portugal, Sept. 1995, 149-161.

[4] Borges, M. R. S., Brézillon, P., Pino, J. A., Pomerol, J.-Ch. Dealing with the effects of context mismatch in group work. In Decision Support Systems, accepted for publication.

[5] Borges, M. R. S., Brézillon, P., Pino, J. A. Bringing Context to CSCW. In Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design. CSCWD´2004, Xiamen, China, May 2004, 161-166.

[6] Borges, M. R. S., Pino, J. A. Awareness Mechanisms for Coordination in Asynchronous CSCW. In Proc. 9th Workshop on Information Technologies and Systems (WITS 1999), Charlotte, North Carolina, Dec. 1999, 69-74.

[7] Brézillon, P. Context in problem solving: A survey. In The Knowledge Engineering Review, 1999, 14(1), 1-34.

[8] Brézillon, P. Making context explicit in communicating objects. In Kintzig, C., Poulain, G., Privat, G., Favennec, P.-N. (eds), Communicating with Smart Objects, London: Kogan Page Science, 273-284.

[9] Brézillon, P., Borges, M. R. S., Pino, J. A., Pomerol, J.-Ch. Context-Awareness in Group Work: Three Case Studies. In Proceedings of the IFIP International Conference on Decision Support Systems, Decision Support in an Uncertain and Complex Word, 2004, v.1, 115-124.

[10] Borges, M. R. S., Pino, J. A., Fuller, D. A., Salgado, A. C. Key issues in the design of an asynchronous system to support meeting preparation. Decision Support Systems – The International Journal, 27, 1999, 269-287.

[11] Bostrom, R. P., Anson, R., Clawson, V. K. Group Facilitation and Group Support Systems. In Jessup, L. M., Valacich, J. S. (eds), Group Support Systems – New Perspectives, chapter 8, Macmillan Publishing Company, New York, Maxwell Macmillan Canada, Toronto, Maxwell Macmillan Int., New York, Oxford, Singapore, Sydney, 1993, 146-168.

[12] Brézillon, P., Borges, M. R. S., Pino, J. A., Pomerol, J.-Ch. Context-Awareness in Group Work: Three case Studies. In Proc. of the 2004 IFIP Int. Conf. on Decision Support Systems, Prato, Italy, 2004, 115-124.

[13] David, J. M. N., Rosa, M. G. P., Santoro, F. M., Borges, M. R. S. Considering Context Elements in Pre-meeting Systems. In Proc. of the 10th International Conference on CSCW in Design, ISBN: 1-4244-0165-8, Southeast University, Nanjing, China, May 2006, 209-214.

[14] David, J. M. N., Borges, M. R. S. Designing collaboration through a web-based groupware infrastructure. In International Journal of Computer Applications in Technology, v. 19, n. 3/4, ISSN 0952-8091, Inderscience, 2004, 175-183.

[15] Dey, A. K., Salber, D., Abowd, G. D. A Conceptual Framework and Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In Human-Computer Interaction Journal, 2001, v. 16(2-4), 97-166.

[16] Dourish, P., Bellotti, V. Awareness and Coordination in Shared Workspaces. In Proc. ACM Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, Canada, Nov. 1992, 107-114.

[17] Greenberg, S. Context as a Dynamic Construct. In Human-Computer Interaction, Volume 16 (2-4), Lawrence Erlbaum Associates Inc., 2001, 257-268.

[18] Guerrero, L. A., Pino, J. A. Preparing Decision Meetings at a Large Organization. In Proc. of the International Conference on Decision Making and Decision Support in the Internet Age (DSIAge'2002), Cork, Ireland, Jul. 2002, 85-95.

[19] Pinheiro, M. K., De Lima, J. V., Borges, M. R. S. Awareness em Sistemas de Groupware. In Proc. of the IDEAS 2001, San Jose, Costa Rica, Abr. 2001, 323-335.

[20] Rittenbruch, M. ATMOSPHERE: A Framework for Contextual Awareness. In International Journal of Human-Computer Interaction, v. 14(2), 2002, 159-180.

[21] Rosa, M. G. P., Borges, M. R. S., Santoro, F. M. A Conceptual Framework for Analyzing the Use of Context in Groupware. In Favela, J., Decouchant, D. (eds), Groupware: Design, Implementation, and Use. Proceedings of 9th International Workshop on Groupware (CRIWG'03), Lecture Notes in Computer Science, v. 2806, ISBN 3-540-20117-3, Springer-Verlag, Autrans, France, Sept. 28 - Oct. 2. 2003, 300-313.

[22] Salvador, T., Scholtz, J., Larson, J. The Denver Model for Groupware Design, SIGCHI Bulletin, Vol. 28, No.1, Jan. 1996, 52-58.

[23] Santoro, F. M., Borges, M. R. S., Santos, N. Um framework para estudo de ambientes de suporte à aprendizagem cooperativa. In Revista Brasileira de Informática na Educação, n. 04, 1999, 51-68.

[24] Siebra, S. A., Salgado, A. C., Tedesco, P. A., Brézillon, P. A Learning Interaction Memory using Contextual Information. In Fifth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT´05), Workshop on Context and Groupware. CEUR Proceedings, Vol. 133, Paris, France, July 2005.

[25] Vieira, V., Tedesco, P., Salgado, A. C. Towards an Ontology for Context Representation in Groupware. In Fuks, H., Lukosch, S., Salgado, A. C., (eds.), Groupware: Design, Implementation, and Use. Proceedings of 11th International Workshop on Groupware (CRIWG'05), Lecture Notes in Computer Science, v. 3706, ISBN 3-540-29110-5, Springer-Verlag, Porto de Galinhas, Brazil, Sept. 25 to 29, 2005, 367-375.

Um Sistema de Recomendação de Especialistas Sensível ao Contexto para Apoio à Colaboração Informal

Helô Petry
CIn/UFPE
C.P. 7851, Recife-PE
+55-81-21268430
hp@cin.ufpe.br

Vaninha Vieira
CIn/UFPE
C.P. 7851, Recife-PE
+55-81-21268430
vvs@cin.ufpe.br

Patricia Tedesco
CIn/UFPE
C.P. 7851, Recife-PE
+55-81-21268430
pcart@cin.ufpe.br

Ana Carolina Salgado
CIn/UFPE
C.P. 7851, Recife-PE
+55-81-21268430
acs@cin.ufpe.br

RESUMO

Freqüentemente, pessoas precisam, para realização de sua tarefa, de determinado conhecimento que só pode ser obtido através de experiência e prática, que elas podem não possuir. Essas pessoas podem economizar tempo e esforço se puderem interagir, informalmente, com outras pessoas que já possuam esse conhecimento. Um desafio da colaboração informal é justamente identificar esses possíveis colaboradores, em função de seus interesses, deficiências e contexto atual de trabalho. Este artigo apresenta o ICARE (*Intelligent Context Awareness for Recommending Experts*) um sistema de recomendação de especialistas que utiliza o contexto do usuário e do especialista para fazer recomendações mais ajustadas às necessidades do usuário e, dessa maneira, facilitar a colaboração informal.

ABSTRACT

Frequently, to better perform their tasks, people need knowledge that can only be obtained through hands-on experience, which they might not possess. Such people can save time and effort if they can informally interact with others that have such knowledge. A challenge in informal collaboration is to identify these potential collaborators, taking into account their interests, deficiencies and current work context. In this light, this paper presents ICARE (Intelligent Context Awareness for Recommending Experts), an expert recommendation system that considers the context of both the user and the expert, in order to provide recommendations that fit the users' needs better and, in this way, enhance the informal collaboration.

Palavras Chave

Computação ciente de contexto, sistemas de recomendação, colaboração informal, colaboração contextual.

Keywords

Context-aware computing, recommendation systems, opportunistic collaboration, contextual collaboration

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 38-47.

1. INTRODUÇÃO

A crescente competitividade e exigência de lançamentos em menos tempo de produtos cada vez melhores têm exigido que as organizações otimizem seus processos de trabalho e passem a contar cada vez mais com equipes trabalhando colaborativamente para desenvolvimento das tarefas.

Uma questão importante quando se pensa em trabalho colaborativo é como prover meios que melhorem a interação e colaboração entre os indivíduos. Para que as atividades do grupo sejam efetivas é necessário que seja viabilizada a comunicação, o compartilhamento e o reuso de conhecimento entre os indivíduos [11].

Reusar conhecimento está fortemente relacionado a explorar melhor o capital humano existente dentro do grupo ou organização, de modo a evitar gastos de tempo e dinheiro na aquisição do conhecimento cada vez que ele for necessário. Porém, o conhecimento nem sempre é algo facilmente acessível, visto que ele pode estar armazenado como conhecimento tácito, desorganizado dentro da cabeça dos indivíduos, e sua elicitação pode não ser trivial [6].

Muitas vezes, o indivíduo se depara com tarefas que demandam conhecimento de um assunto que este não possui. Nessas situações pode haver ganho de tempo se ele puder interagir com algum especialista que o auxilie em um primeiro momento. No entanto, pode ser bastante complexo identificar quem poderia ser esse especialista, onde está e se está disponível para colaborar com o indivíduo.

Um sistema que perceba esses potenciais colaboradores e os coloque em contato trará um grande benefício ao trabalho desenvolvido por ambos. Afinal, a comunicação direta torna muito mais fácil e efetivo o compartilhamento dos conhecimentos tácitos e habilidades entre os indivíduos.

Sistemas de Recomendação de Especialistas (SRE) são sistemas desenvolvidos para produzir capital social e humano em uma organização [10]. Eles podem ser vistos como uma tentativa de oferecer acesso ao conhecimento implícito, difícil de ser exteriorizado. SRE retornam referências para atores humanos que são identificados como "especialistas" em um domínio requerido [10].

SRE podem ser usados para apoiar a colaboração informal através da identificação de especialistas que possam ajudar indivíduos na realização de suas tarefas. No entanto, para que seja efetivo, o SRE precisa conhecer o contexto do especialista que será recomendado, bem como do usuário que receberá a

recomendação, de modo a buscar dentre os especialistas existentes os mais viáveis para a situação atual do usuário. Outra funcionalidade interessante é a identificação pró-ativa da necessidade de recomendar um especialista a um usuário, com base em seu contexto.

Este artigo apresenta a proposta de um SRE, denominado ICARE (*Intelligent Context Awareness for Recommending Experts*). O objetivo desse sistema é apoiar a etapa inicial de uma colaboração informal, ou seja, identificar com quem o usuário pode interagir e como iniciar essa interação. Para isso, o ICARE recomenda especialistas de acordo com assuntos relacionados à tarefa do usuário.

O ICARE é um SRE sensível ao contexto dos especialistas e dos usuários que receberão a recomendação. Ele usa esse conhecimento contextual para fazer recomendações de acordo com as necessidades e disponibilidades atuais desses indivíduos. Os especialistas que podem ser recomendados são classificados e apresentados, preferencialmente, em função do que será mais útil para o usuário no momento, como por exemplo, a reputação do especialista, sua disponibilidade, o nível de especialidade, a proximidade social, ou a proximidade física.

O conhecimento contextual permite que o ICARE faça recomendações mais adequadas e viáveis a uma colaboração imediata. Por exemplo, possivelmente não é útil para um usuário que precise criar um diagrama UML em seu projeto, e que esteja com problemas no processo RUP, receber a recomendação de um especialista em RUP que também é gerente de um projeto com prazo esgotando em dois dias.

O restante deste artigo está organizado como segue: na Seção 2, serão discutidos aspectos relacionados à colaboração informal; a Seção 3 expõe o conceito de contexto e como o mesmo pode ser utilizado para melhorar a colaboração; na Seção 4 são discutidos os desafios subjacentes aos SRE e são apresentados alguns SRE existentes na literatura; a Seção 5 descreve a proposta do ICARE, sua arquitetura e componentes principais; a Seção 6 discute um exemplo de uso; e, finalmente, a Seção 7 apresenta as considerações finais e direções para trabalhos futuros.

2. COLABORAÇÃO INFORMAL

Uma atividade colaborativa pode ocorrer de diversas maneiras. As pessoas podem agendar reuniões para um trabalho planejado, podem ser alocadas para tarefas compartilhadas ou podem simplesmente interagir informalmente quando uma oportunidade para um trabalho compartilhado se apresente [5]. Colaboração informal são interações não planejadas e que ocorrem de forma oportunista.

Muitas afinidades e oportunidades para colaboração e troca de conhecimento são descobertas em interações que ocorrem informalmente, como em conversas na sala do cafezinho da empresa, ou em *happy hours* no final do expediente. Um exemplo é o gerente de uma equipe de desenvolvimento de software que recebe a solicitação de um novo projeto que exige conhecimento sobre construção de navios. Em conversas informais na empresa ele expõe para colegas essa situação para ver se alguém conhece uma pessoa que tenha essa especialidade. Durante uma dessas conversas ele se depara com um colega de outro projeto, que já trabalhou no desenvolvimento de software para suporte à construção de navios.

Gutwin et al. apontam algumas observações relacionadas à colaboração informal [5]: (i) são comuns em grupos co-localizados; (ii) em geral são breves e freqüentes; (iii) se baseiam na percepção dos indivíduos sobre seu ambiente de trabalho; (iv) podem ser iniciadas pela percepção de pessoas, objetos, ações ou outras interações; (v) ocorrem, geralmente, de forma pensada, de acordo com a necessidade dos participantes; (vi) são fáceis de iniciar; e (vii) exigem, porém, um processo de vários passos para que alguém se engaje na interação (e.g. observar disponibilidade e/ou receptividade).

Como podemos observar, existem dois aspectos cruciais para viabilizar a colaboração informal. Primeiro, é necessária uma percepção sobre o ambiente de trabalho para identificar indivíduos com os quais se possa colaborar. E, segundo, é importante verificar a viabilidade dessa interação, seja em termos da disponibilidade dos envolvidos ou da receptividade dos indivíduos para essa interação.

No entanto, esses potenciais colaboradores podem não estar fisicamente próximos, de modo a se esbarrar pelos corredores da empresa, e podem não possuir *a priori* conhecimento que permita identificar essa possibilidade de interação. Dessa forma, faz-se necessário um apoio computacional que viabilize o estabelecimento desse tipo de colaboração, ou seja, que coloque os dois indivíduos em contato.

Para apoiar a colaboração informal, o sistema deve ter conhecimento sobre informações relacionadas ao contexto atual das *pessoas*, como: onde estão; o que estão fazendo; presença; disponibilidade; interesses; e habilidades. Informações sobre artefatos em uso e sobre o ambiente computacional também são úteis, pois a observação de um artefato, documento, ou software em execução por um usuário pode conduzir a uma interação. Por exemplo, João está começando a trabalhar com ontologias e sabe que o Protégé é uma ferramenta para edição de ontologias, porém está com dificuldades em sua utilização. Se João percebe que Paulo trabalha ou está usando o Protégé, essa informação pode ajudá-lo a ver em Paulo uma pessoa com quem pode interagir para tirar dúvidas sobre o trabalho com a ferramenta.

Percebe-se que o conhecimento contextual é fundamental para viabilizar a identificação e estabelecimento de colaborações informais. A próxima seção discute o que é contexto e o seu papel em sistemas colaborativos de modo geral.

3. CONTEXTO NA COLABORAÇÃO

Nas mais diversas situações do dia-a-dia as pessoas utilizam conhecimento do contexto para delimitar e direcionar suas ações e comportamentos. As mensagens trocadas para comunicação trazem junto um contexto associado que apóia a compreensão do seu conteúdo. Contexto ajuda a melhorar a qualidade da conversação e a compreender certas situações, ações ou eventos. Contexto é o que está por trás da habilidade de discriminar o que é ou não é importante e, com isso, permite que os indivíduos gerenciem a grande quantidade de informação que os cerca em um dado instante.

Contexto é uma importante ferramenta para apoiar a comunicação entre pessoas e sistemas computacionais, pois ajuda a diminuir ambigüidades e conflitos, aumenta a expressividade dos diálogos, e possibilita a melhoria dos serviços e informações oferecidos pela aplicação. Sistemas sensíveis a contexto são aqueles capazes de entender o contexto

dos usuários e antecipar suas necessidades, em termos de serviços e/ou informações relevantes [4].

Bazire e Brézillon coletaram um conjunto de aproximadamente 150 definições de contexto, vindas de diferentes domínios e concluíram que o contexto atua como um conjunto de restrições que influenciam o comportamento de um sistema embutido em uma dada tarefa e que a definição de contexto depende da área de conhecimento à qual pertence [1].

O uso do contexto é bem conhecido e amplamente propagado em aplicações de computação ubíqua [3]. No entanto, existe um interesse crescente em utilizar esse conceito em outras áreas, como sistemas colaborativos para facilitar o entendimento e cooperação entre os membros do grupo, e ajudar a melhorar a coesão e eficiência do grupo [2]. Nesses sistemas, o conceito de contexto já está presente, embora associado a tópicos como percepção e memória do grupo [2].

As informações contextuais relevantes para sistemas colaborativos diferem bastante do seu uso em computação ubíqua. Nesse último, o contexto se baseia muito em informações que podem ser adquiridas por meio de sensores físicos, como localização, dispositivos e informações sobre o ambiente físico. Em sistemas colaborativos, as informações contextuais de interesse são relativas ao ambiente virtual de trabalho, organização do grupo, características e papéis dos seus membros, artefatos compartilhados, divisão de tarefas, prazos, objetivos, entre outras.

No entanto, essas informações, ao contrário das adquiridas por sensores físicos, não possuem padrões para sua representação. Visando estruturar informações de contexto relacionadas a esses ambientes, Vieira et al. propuseram uma ontologia para representação de contexto em sistemas colaborativos [12]. Essa ontologia classifica o contexto em três classes principais: *contexto organizacional*, que representa a informação contextual relativa a toda a estrutura organizacional do grupo, como seus membros, o papel que cada um desempenha na execução de uma tarefa, em um projeto específico; *contexto da interação*, que indica as informações referentes ao contexto de uma interação que está acontecendo (síncrona) ou que já ocorreu (assíncrona), em termos dos estados dos artefatos compartilhados e das ferramentas utilizadas; e *contexto físico*, que contém informações sobre os elementos físicos que caracterizam o ambiente do usuário.

Devido à natureza da informação contextual, a aquisição do contexto em sistemas colaborativos exige uma maior participação do usuário no fornecimento de algumas dessas informações. Essa demanda extra pode desmotivar o usuário a utilizar a ferramenta ou pode levá-lo a preencher de forma errônea ou incompleta. Os benefícios no uso do contexto devem, pois, ser percebidos logo pelo usuário para motivá-lo a continuar alimentando o sistema.

O conhecimento do contexto dos usuários é fundamental para viabilizar a colaboração informal, tanto para identificar os indivíduos com os quais se pode colaborar, quanto para verificar o melhor momento para solicitar essa interação. A próxima seção discute os conceitos associados a sistemas de recomendação de especialistas.

4. SISTEMAS DE RECOMENDAÇÃO DE ESPECIALISTAS

Um sistema de recomendação de especialistas (SRE) é um sistema que retorna referências para atores humanos que são identificados como "especialistas" no domínio requisitado [10]. Uma vez que conhecimento, em termos de experiência, habilidades ou interesses, está ligado de forma inerente a atores humanos, SRE têm surgido, focando na conexão entre pessoas que precisam trocar informações.

Existem diversos SRE desenvolvidos (e.g. [7-10]). Eles utilizam diversas técnicas de construção de perfis, abordagens de recomendação e critérios para determinar se uma pessoa é um especialista e em qual área do conhecimento. Mas nenhum desses sistemas leva em consideração o contexto do usuário que receberá a recomendação ou do especialista possivelmente recomendado. Acreditamos que um SRE sensível a contexto pode ser mais eficiente e inteligente, pois pode levar em consideração o contexto do usuário e, portanto, fazer recomendações mais direcionadas às suas necessidades.

Um dos desafios enfrentados na recomendação de especialistas é exatamente a definição do que considerar como especialista. Ou seja, o que faz uma pessoa ser considerada um especialista? Que informações são necessárias para se afirmar que uma pessoa seja um especialista em determinada área do conhecimento? O que difere um especialista em determinado campo de conhecimento de uma pessoa leiga nesse campo? Esse problema se reflete nos SRE existentes. Cada sistema utiliza diferentes indicativos de especialidade e diferentes fontes dessa informação.

Por exemplo, a depender do Indicativo de Especialidade (IE) considerado, um especialista em Ornitorrincologia[1] é:

1. Um autor freqüente de artigos científicos sobre Ornitorrincos (IE = publicações);
2. Um orientador experiente de trabalhos acadêmicos em Ornitorrincologia, como dissertações de mestrado e teses de doutorado (IE = orientações acadêmicas);
3. Um profissional experiente em Ornitorrincologia (IE = tempo de experiência);
4. Uma pessoa com vários certificados em Ornitorrincologia (IE = certificações);
5. Um profissional que já participou de inúmeros projetos em Ornitorrincologia e, de preferência, projetos bem sucedidos (IE = participação em projetos);
6. Uma pessoa que participa bastante de fóruns on-line e costuma responder a dúvidas sobre Ornitorrincologia postadas nesses fóruns (IE = participação em fóruns).

Fica evidente que a melhor definição de um especialista vai depender de quem está buscando o especialista e qual a tarefa que essa pessoa tem em mãos. Por exemplo, se o SRE está sendo usado por alguém que precisa escrever um artigo sobre Web Semântica, a melhor definição de especialista para esse usuário é a primeira definição dada acima (autor experiente). Já um programador que pede ao SRE um especialista em programação distribuída pode preferir um especialista caracterizado pela 4ª definição (com certificados). Mas os SRE

[1] Área de conhecimento fictícia usada como exemplo

existentes atualmente não possuem essa flexibilidade. São construídos levando em consideração uma definição e não permitem que o usuário escolha o tipo de especialista que melhor lhe convém.

Outro desafio dos SRE é saber onde procurar pelos especialistas. Algumas possibilidades são: (i) páginas de publicações científicas (CiteSeer[2], por exemplo); (ii) fóruns na Internet; (iii) páginas amarelas da organização/empresa, se disponível.

Também podemos perceber que essas fontes de especialistas mudam de acordo com a definição de especialista utilizada. O ideal seria que a fonte do SRE também pudesse ser variada, de acordo com a necessidade atual do usuário. Novamente, os SRE atuais não atendem essa necessidade, já que mantêm fixa a origem dos especialistas recomendados.

Essas fontes são utilizadas para gerar um perfil do usuário, ou seja, definir quais áreas do conhecimento representam as especialidades ou interesses do usuário. Reichling et al. [10] apresentam três classes de SRE de acordo com os tipos de dados utilizados na geração do perfil do usuário:

1. *Dados históricos:* lista dos endereços favoritos do usuário ou histórico de navegação são fontes de dados históricos do usuário que possivelmente representam os interesses ou competências do usuário.
2. *Documentos*: documentos que são escritos, lidos ou revisados pelo usuário parecem ser outra fonte de dados indicativa dos interesses do usuário. 'Documentos' incluem documentos de texto, assim como e-mails ou mensagens para grupos de discussão.
3. *Rede social:* analisar o ambiente social do usuário é outro método de avaliar o usuário. A idéia básica é que usuários que colaboraram no passado possivelmente colaborarão com sucesso no futuro.

O terceiro desafio dos SRE é a apresentação ao usuário dos especialistas na área de interesse encontrados. Afinal, a ordem dos resultados mostrados é importante. O usuário dificilmente estará disposto a avaliar **todos** os especialistas encontrados. Estamos acostumados a receber resultados de consultas ordenados por relevância. Portanto, é necessário dar prioridade aos especialistas mais recomendáveis encontrados, ou seja, aqueles que mais se adequam às necessidades do usuário. Esse *ranking* dos especialistas obtidos também pode ser feito de acordo com diversos critérios, como:

- Grau de especialidade, se disponível. Caso o indicativo de especialidade utilizado possibilite uma quantificação, além de uma classificação simples: especialista e não especialista.
- Reputação do especialista, calculada em função de recomendações anteriores e do grau de especialidade. Ou seja, outra pessoa no passado também procurou um Ornitorrincólogo e recebeu como resultado uma referência a José da Silva. O usuário entrou em contato com José da Silva, tirou todas suas dúvidas e depois afirmou ao SRE que recomendaria o especialista José da Silva. Assim, a reputação do especialista José da Silva é melhorada.
- Proximidade. Caso a informação de localização das pessoas envolvidas seja disponível, o SRE pode dar mais importância para especialistas fisicamente próximos do usuário. Afinal, um contato físico pode ser mais efetivo.
- Disponibilidade. Se o SRE puder informar se um especialista está ocupado (ou quão ocupado), pode dar a opção de o usuário abordar apenas alguém que tenha mais tempo para colaborar.
- Rede social. O SRE prioriza especialistas que conhecem algum amigo ou conhecido do usuário, facilitando a abordagem ao especialista.

4.1 Trabalhos Relacionados

Alguns SRE existentes na literatura são discutidos a seguir.

O *TABUMA* (*Text Analysis Based User Matching Algorithm*) [10] é uma instância do *Framework ExpertFinding* e utiliza documentos de texto fornecidos pelo usuário para estimar suas preferências. O sistema gera perfis de usuários a partir de documentos de texto através de métodos lingüísticos para extração de palavras-chave. Dois passos são realizados assincronamente: (1) criação do perfil a partir de documentos diversos e (2) *matching* dos perfis entre si. O passo 1 é uma ação realizada perifericamente para manter os perfis dos usuários atualizados. O passo 2 é executado por um pedido do usuário. Um ponto forte desse sistema é a flexibilidade obtida com a liberdade de fornecer qualquer tipo de documento de texto. Um ponto negativo é a necessidade de o usuário deliberadamente escolher um conjunto de documentos que ele considere representante do seu conhecimento. Além do custo para o usuário realizar essa tarefa, esse método pode não ser completamente confiável.

O Sistema HALe [8] encontra conexões (implícitas e explícitas) entre pessoas a partir das suas comunicações por e-mail. Relações entre os usuários e tópicos de interesse também são extraídos dos textos dos e-mails. São utilizadas técnicas lingüísticas para minerar essas associações. Portanto, esse sistema considera especialista uma pessoa que escreve muitos e-mails sobre o assunto. O ponto forte do HALe é sua transparência para o usuário, que não tem suas atividades perturbadas pelo sistema. O ponto negativo é a quebra da privacidade do usuário, já que seus e-mails são monitorados.

O *Expertise Recommender* (ER) [9] baseia-se na idéia de que relacionamentos são um fator chave para cooperação dentro de organizações. O sistema filtra suas recomendações de acordo com a rede social do requerente. De maneira simplificada, o processo de recomendação do ER é composto de dois passos. Primeiro, o ER encontra um conjunto de indivíduos que provavelmente têm a especialidade necessária. Então, essas potenciais recomendações são combinadas com a pessoa (que está procurando a especialidade) usando uma rede social. Durante a avaliação do sistema, percebeu-se um *trade-off* entre recomendar a pessoa mais capacitada ou a pessoa com a qual se pode interagir mais facilmente. O ponto forte do ER é considerar o contexto social do usuário e ser flexível quanto aos indicativos de especialidade utilizados e suas fontes de dados. O primeiro ponto negativo do ER é utilizar técnicas etnográficas (caras e demoradas) para levantar as redes sociais existentes na organização. O segundo ponto negativo do ER é a necessidade de configuração de acordo com uma heurística específica de

[2] http://citeseer.ist.psu.edu

domínio usada para caracterizar um especialista. Por exemplo, em uma empresa específica de desenvolvimento de software, o especialista em determinado código produzido no passado é o programador que fez a última alteração no código. Nesse caso, o sistema é configurado para acessar o banco de dados que registra alterações feitas nos artefatos (assim como o autor e data das alterações) e obter o nome do funcionário responsável pela última alteração.

O *ReferralWeb* [7] utiliza a rede social do ator para fazer recomendações. O objetivo é reconhecer comunidades baseadas na co-autoria de artigos científicos. *ReferralWeb* considera como evidência de relacionamento direto entre duas pessoas a co-ocorrência de seus nomes, localizados próximos um do outro, em documentos públicos disponíveis na Web. Esses documentos usados como fontes incluem listas de co-autores em artigos técnicos e citações de artigos. *ReferralWeb* assume, por exemplo, que o nível de especialidade de uma pessoa em um determinado tópico varia de forma monotônica com o número de documentos em que as palavras-chave e a pessoa aparecem. Portanto, um especialista em um assunto é considerado aquele cujo nome aparece em muitas páginas de Internet junto com o assunto em questão. Para localizar documentos relevantes para uma pessoa, o *ReferralWeb* usa um motor de busca genérico (AltaVista). Um ponto forte do *ReferralWeb* é a identificação de grupos de especialistas e o grau dos relacionamentos entre esses especialistas. Mas o *ReferralWeb* não possui um mecanismo de busca específico para especialistas.

A Tabela 1 resume as principais características dos SRE apresentados, comparando as técnicas utilizadas para construir os perfis dos usuários e que tipo de informação é considerada para indicar que uma pessoa é um especialista.

Tabela 1 Sistemas de Recomendação de Especialistas

Ferramenta	Indicador de especialidade	Técnica para construção do perfil do usuário
TABUMA	Arquivos de texto não-estruturados fornecidos pelo usuário	Métodos automáticos de análise de texto
HALe	Volume de e-mails enviados sobre o assunto	Análise lingüística de e-mails
Expertise Recommender	Heurística específica do domínio. Ex: última pessoa que alterou um código	Arquitetura deixa a questão em aberto. Ex: minerar BD em busca de registros de alterações de código
ReferralWeb	Número de páginas em que as palavras-chave e a pessoa aparecem	Mineração de dados na Web

5. ICARE: UM SRE SENSÍVEL A CONTEXTO

Avaliando os SRE existentes, verificamos a necessidade de maior flexibilidade na definição de especialista pelos sistemas. Cada sistema apresentado tem uma visão de especialista definida *a priori* e é construído dirigido por essa definição. Ou seja, um especialista é quem lê muito sobre o assunto, portanto tem vários documentos sobre esse assunto. Ou um especialista é quem publica artigos sobre o assunto, portanto aparece em várias páginas da Internet, inclusive páginas com listas de publicações.

Neste trabalho, propomos o ICARE (*Intelligent Context Awareness for Recommending Experts*). ICARE é um SRE que utiliza o contexto de trabalho do usuário que receberá a recomendação e do especialista a ser recomendado, se possível (caso o sistema responsável pela aquisição das informações de contexto esteja instalado na máquina do especialista). O sistema utiliza o contexto do usuário (tarefa em que está envolvido, perfil de interesses e habilidades) para encontrar uma definição de especialista mais adequada para este contexto. Em função da definição de especialista, a escolha das fontes de informações sobre estes é flexibilizada. O contexto também é usado para ajudar no *ranking* dos resultados encontrados: dar prioridade aos especialistas que melhor atendam às necessidades do usuário no contexto atual.

Para obter as informações de contexto necessárias para sua atividade, o ICARE utiliza os serviços de aquisição, análise e gerenciamento de contexto providos pelo CoMGroup (do inglês *Context Manager for Groupware*) [12]. O CoMGroup é um gerenciador de contextos responsável por perceber o foco de trabalho corrente do usuário e prover mecanismos de raciocínio e inferência que permitam identificar qual o contexto correspondente a esse foco, a partir da manutenção de uma base de conhecimento contextual. Essa ferramenta é acoplada ao ambiente computacional de um usuário para obter seu contexto. ICARE e CoMGroup possuem uma API de comunicação, como ilustra a Figura 1, através da qual o ICARE solicita e recebe as informações de contexto do usuário monitorado pelo CoMGroup. Portanto, para considerar o contexto de um usuário ou especialista, este precisa possuir o CoMGroup instalado em seu ambiente computacional.

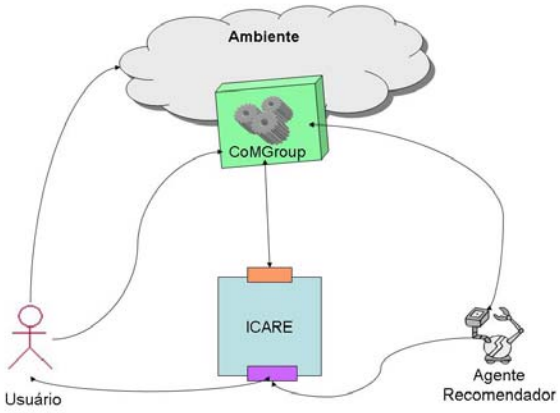


Figura 1 Aquisição de Contexto pelo ICARE

O ICARE possui mais uma interface de comunicação, a Interface de Recomendação (Figura 1), por onde recebe as solicitações de recomendação e as especialidades requeridas. Essa Interface pode ser utilizada diretamente pelo usuário,

através de um módulo de interface gráfica que apenas recebe as palavras-chave digitadas e as repassa para o ICARE, ou por um Agente Recomendador. Esse agente obtém o contexto do usuário através do CoMGroup e decide quando fazer uma recomendação e em que especialidade.

As possíveis informações contextuais obtidas pelo ICARE variam de acordo com o ambiente computacional e com as fontes de contexto cadastradas pelo CoMGroup do usuário. Alguns exemplos de informações de contexto possivelmente utilizadas pelo ICARE são: (i) o calendário ou agenda do usuário (e do especialista, se possível); sabendo das atividades planejadas das pessoas, o sistema pode inferir a disponibilidade delas; (ii) ferramenta(s) sendo usada(s) no momento, como navegador, editor de texto, ferramenta de desenvolvimento de software, visualizador de pdf; de posse dessas informações o ICARE pode inferir a atividade atual do usuário, como: lendo, escrevendo ou respondendo e-mails; (iii) perfil do usuário (e.g. desenvolvedor, estudante, professor, gerente); ajuda a identificar que tipo de especialista é mais apropriado; (iv) interesses/ habilidades; serve para alinhar o conhecimento do especialista recomendado com o do usuário, além de possibilitar que o usuário recebedor da recomendação também possa ser um especialista a ser recomendado.

A arquitetura do sistema, como mostra a Figura 2 é dividida em quatro módulos (que serão descritos em detalhes nas subseções a seguir), cada um responsável por uma etapa da recomendação de especialistas: *Determinador de Especialidade*; *Recuperador de Especialistas*; *Apresentador de Especialistas* e *Gerador da Base de Especialistas*.

5.1 Determinador de Especialidade

A primeira etapa no processo de recomendação de especialista é descobrir qual é a especialidade que se deverá buscar, ou seja, definir a área do conhecimento humano em que se buscará um especialista. Essa etapa recebe como entrada algumas palavras-chave que caracterizam a especialidade desejada e devolve um conceito ontológico relativo à especialidade classificada. O Determinador de Especialidade dispõe de uma ontologia de domínio cuja finalidade é uniformizar os conceitos utilizados para definir as especialidades. Essa ontologia expressa conceitualizações das áreas e subáreas de um domínio de conhecimento.

Esse módulo possui um agente inteligente responsável por obter o contexto relevante do usuário. Para isso, ele se comunica com o CoMGroup e descobre que o usuário está navegando em fóruns sobre Oracle Lite, por exemplo. De posse do contexto do usuário e (possivelmente) analisando bases de casos de recomendação anteriores <contexto do usuário, recomendação de especialista feita, avaliação da recomendação pelo usuário>, o Determinador de Especialidade estima a especialidade que melhor atende às necessidades do usuário no contexto atual. Ex: bancos de dados móveis e Oracle.

5.2 Recuperador de Especialistas

Conhecendo-se a especialidade requerida, o próximo passo é buscar um especialista correspondente. Ou seja, descobrir as pessoas que possivelmente são especialistas no assunto requerido. Portanto, o Recuperador de Especialistas é responsável por varrer uma lista de pessoas cujas especialidades são conhecidas (ou calculadas ou inferidas pelo sistema) e identificar aquelas especialistas no assunto em questão.

Essa busca envolve alguns sub-problemas. Primeiro, como definir um especialista. Como discutido anteriormente, a melhor definição vai depender do contexto do usuário: em qual atividade está envolvido e em que o especialista irá colaborar. Assim, a etapa de recuperação de especialistas utiliza duas informações de contexto do usuário: seu perfil e atividade atual.

O ICARE utiliza uma função de utilidade que leva em consideração todas essas possíveis definições, dando pesos diferentes para cada indicativo de especialidade. De acordo com o contexto do usuário, determinados indicativos terão mais peso que outros. Por exemplo, se o contexto do usuário é "escrevendo artigo", o peso do indicativo "número de publicações" será maior.

O segundo problema do Recuperador de Especialistas é definir que fontes de especialistas utilizar. Algumas possibilidades são: páginas de publicações científicas; fóruns na Internet; páginas amarelas da organização/empresa; usuários cadastrados no ICARE (alguém que no passado recebeu uma recomendação no mesmo assunto pode ter bons conhecimentos no assunto). Novamente, aqui a melhor fonte de informação vai depender do contexto e dos indicativos de especialidade utilizados. Por exemplo, se o contexto do usuário é "escrevendo artigo", a fonte de especialista preferencial seriam páginas de publicações científicas. A arquitetura é flexível para aceitar o cadastro de várias e diferentes fontes de especialistas. Mas essas fontes precisam ser previamente cadastradas e os especialistas por elas fornecidos são armazenados na base de dados de especialistas do ICARE. Essa carga é feita pelo Gerador da Base de Especialistas.

Uma vez encontrados os especialistas que serão recomendados, é necessário classificá-los. Afinal, a ordem de apresentação é importante. Portanto, é necessário dar prioridade aos especialistas mais adequados ao usuário (*ranking*).

A classificação dos especialistas pode ser feita de acordo com diversos critérios. Alguns deles são: grau de especialidade (se disponível); reputação (em função de recomendações passadas); proximidade; disponibilidade; rede social (especialista conhece algum amigo do usuário); indicação de um amigo.

5.3 Apresentador de Especialistas

A última etapa da recomendação de especialistas é a apresentação ao usuário dos especialistas descobertos. Nessa etapa já se tem disponível uma lista de especialistas na área de interesse. O objetivo é adequar a apresentação dos especialistas às necessidades, preferências ou contexto do usuário.

O usuário pode preferir receber todos os resultados encontrados e em formato HTML, com links para a página pessoal ou e-mail de cada especialista, por exemplo. Ou o contexto do usuário pode ser "usando PDA" e, portanto, dificilmente ele terá paciência para esperar baixar uma lista muito extensa de especialistas. Assim, a apresentação terá que repassar apenas os primeiros especialistas da lista ordenada recebida.

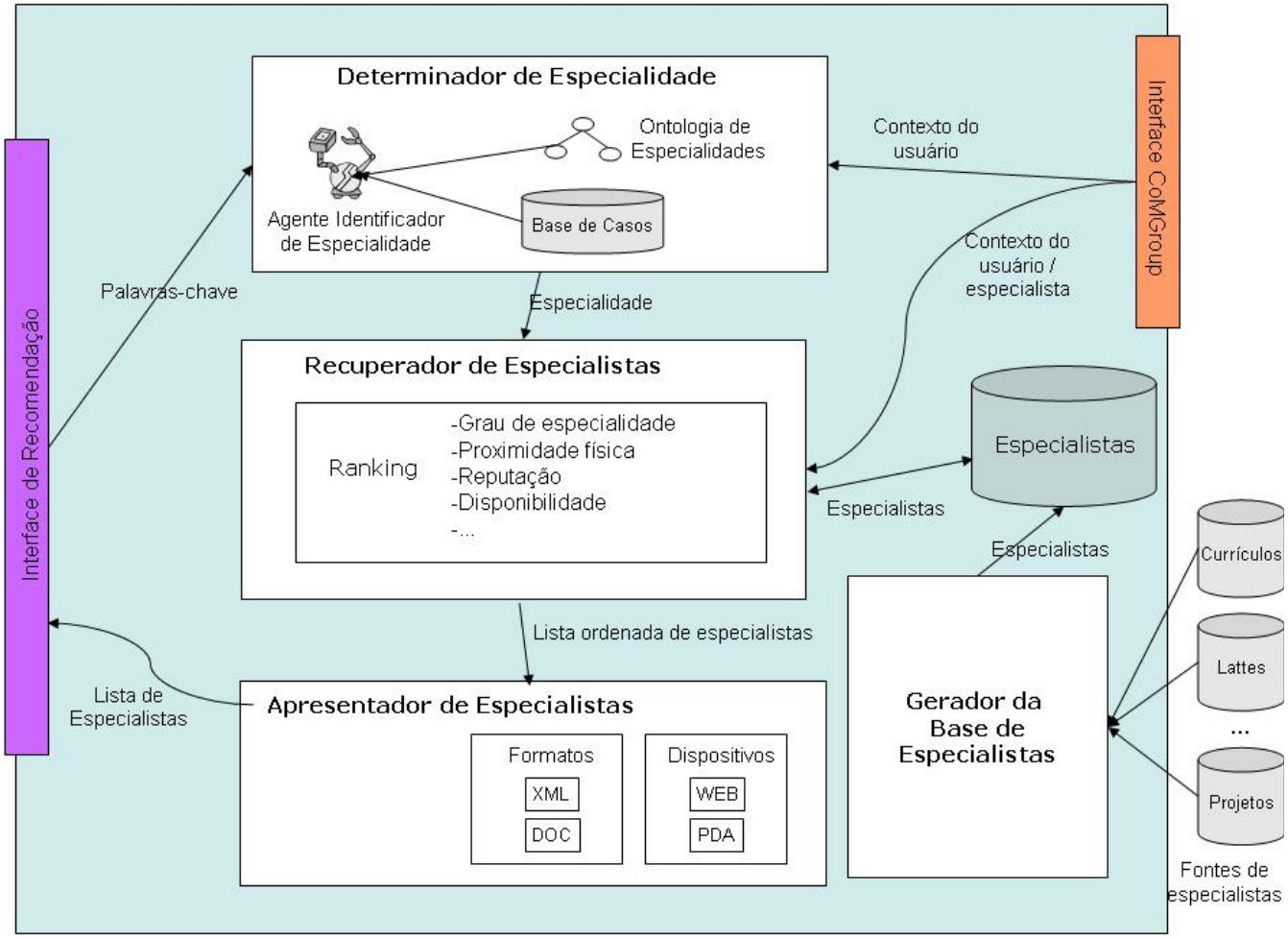


Figura 2 Arquitetura do ICARE

5.4 Gerador da Base de Especialistas

A base de especialistas utilizada pelo Recuperador de Especialistas precisa ser populada. Essa carga é feita previamente à recomendação, de forma que no momento em que um especialista é requerido a base Especialistas já possui todos os dados disponíveis. Periodicamente, o Gerador encarrega-se de atualizar as informações da base de especialistas em função dos dados das fontes.

O Gerador da Base de Especialistas é um componente do ICARE responsável por: (i) cadastrar as fontes de especialistas disponíveis; (ii) receber uma configuração de como obter os especialistas de cada fonte; (iii) guardar na base de dados correspondente as informações dos especialistas obtidas nessas fontes.

Na inclusão de uma fonte de especialistas, o Gerador recebe uma descrição das informações fornecidas pela fonte e como acessá-las. Por exemplo, o script SQL para consultar a base de dados de projetos da companhia.

Portanto, o Gerador tem à disposição todas as fontes cadastradas e está configurado para acessá-las corretamente. Assim, se ele possuir duas fontes, digamos uma base organizacional e um site na Internet como CiteSeer, ele fica apto a utilizar uma fonte ou a outra, ou ambas, para obter os especialistas.

Alguns tipos de especialidades são inerentemente relacionadas à prática de determinada atividade organizacional. Para fazer essa correlação corretamente, o Gerador utiliza um conjunto de regras de produção que associam atividades específicas de uma organização à especialidade correspondente. Por exemplo, um gerente de projeto pode ser associado às especialidades "administração de qualidade de projeto" e "administração de recurso humano de projeto". Essas regras precisam ser inseridas no momento do cadastro da fonte.

6. EXEMPLO DE USO

Nesta seção apresentamos um cenário de uso do ICARE para destacar seus detalhes e funcionalidades.

Seja José um programador inexperiente trabalhando em um projeto (SimpleCode) de desenvolvimento de software realizado na empresa SOFTDEV, que possui 600 funcionários distribuídos em dezenas de projetos. A equipe responsável pelo SimpleCode é composta por 13 pessoas, dentre gerente, líderes de equipe e desenvolvedores. O SimpleCode é desenvolvido em C++ e seus componentes são armazenados em um repositório

com controle de versões (*Concurrent Versions System* - CVS). José é encarregado de codificar um módulo da interface gráfica do sistema e está usando o ambiente de desenvolvimento Visual Studio[3].

Durante a realização da sua tarefa, José precisa utilizar um componente visual específico. Apesar deste ser um componente simples, não está funcionando como previsto pelos manuais de C++. José analisa várias vezes o código, mas não consegue descobrir onde está o erro. Ele utiliza a ajuda do Visual Studio e procura na Internet, mas não encontra uma solução.

As informações de contexto do usuário são obtidos pelo CoMGroup, que monitora o ambiente do usuário. No caso de José, as informações de contexto obtidas estão descritas na Tabela 2.

Tabela 2 Contexto do José

Fonte de contexto	Informação contextual
Ambiente de desenvolvimento de software em execução	Programando
Ambiente Visual Studio em uso	Linguagem C++
Última compilação não foi concluída com sucesso	Problema/dificuldade na programação
"Ajuda" do Visual Studio em uso	Buscando auxílio
Navegando em fóruns de C++ na Internet	Buscando auxílio
O projeto do Visual Studio em execução possui formulário(s)	Interface gráfica
BD do departamento de Recursos Humanos	4 meses de experiência na SOFTDEV; cargo: desenvolvedor júnior
4 meses de experiência	Novato
Cronograma do SimpleCode no Microsoft Project[4]	Prazo de entrega da tarefa atual: 4 dias
Tarefa atual foi programada para ser realizada em um mês	Prazo curto
Login na rede interna	Localização: estação 36
Status no MSN[5]	Disponibilidade: ocupado

O ICARE da SOFTDEV precisa, antes de qualquer coisa, popular sua base de especialistas, utilizando as fontes de especialistas disponíveis (Tabela 3). Essa tarefa é feita pelo **Gerador da Base de Especialistas**, que cadastra as fontes de especialistas. Nesse cadastro, cada fonte é associada ao método utilizado para extração das informações de interesse para o ICARE. Por exemplo, quando a base de projetos da SOFTDEV é incluída, o Gerador é informado como obter (consultas SQL) os dados dos participantes dos projetos, seus cargos e as especialidades envolvidas no projeto.

[3] http://www.msdnbrasil.com.br/visualstudio/default.aspx
[4] http://office.microsoft.com/project
[5] http://www.msn.com

Uma vez informado de como obter as informações necessárias, o Gerador consulta a fonte e replica os dados obtidos na Base de Especialistas, acrescentando o metadado "fonte de origem". Assim o ICARE já pode receber pedidos de recomendações.

Percebendo o conjunto de informações contextuais de José, (programando, usando C++, buscando auxílio), o Agente Recomendador decide recomendar um especialista para ele. Para isso, ele gera uma lista de palavras-chaves que serão utilizadas na busca do especialista e chama o ICARE, passando essa lista de palavras-chave: <C++>.

Tabela 3 Fontes de Especialistas da SOFTDEV

	Fonte de especialistas	Informação obtida
1	Base de dados de projetos realizados e em andamento, com respectivos cronogramas	Participantes envolvidos nos projetos, seus papéis e atividades
2	Base de currículos dos funcionários	Certificações e experiência dos funcionários
3	Base de pessoas do RH	Dados pessoais e de contato
4	Repositórios CVS usados pelas equipes de desenvolvimento	Autores de código e solucionadores de bugs
5	Banco de horas	Atividades realizadas (e em andamento) e seus autores
6	Fórum on-line da SOFTDEV	Autores de dúvidas e soluções
7	Currículo Lattes	Pesquisadores brasileiros
8	CiteSeer	Autores de artigos científicos internacionais

O ICARE recebe a chamada de recomendação, juntamente com a identificação do usuário que receberá a recomendação. O ICARE utiliza a API de comunicação com o CoMGroup de José para obter o seu contexto. O módulo **Determinador de Especialidade** busca na Ontologia de Especialidades por "C++" e encontra um conceito relacionado: <programador C++>. O Determinador de Especialidade também percebe, através da análise de informações de contexto do usuário, que José está trabalhando com interface gráfica. Na Base de Casos existem vários registros de recomendações de especialistas em programação C++ **e** interface gráfica. Assim, o Determinador de Especialidade decide procurar um <programador C++, interface gráfica>. Essa é a especialidade passada para o módulo **Recuperador de Especialistas**.

O Recuperador de Especialistas recebe a especialidade requerida e deve buscar na base de especialistas aqueles que melhor se encaixam com o contexto do usuário (Tabela 2).

Para encontrar os especialistas mais adequados para José, o Recuperador de Especialistas precisa selecionar, dentre as fontes de especialistas disponíveis, quais serão utilizadas. As fontes disponíveis são cadastradas no ICARE e podem ser escolhidas de acordo com o perfil de especialista requerido pelo contexto do usuário.

As fontes de especialistas disponíveis para o ICARE da SOFTDEV são apresentadas na Tabela 3. Considerando o contexto da tarefa e atividade atual de José, o Recuperador de Especialistas seleciona as fontes 1, 4 e 6. O Recuperador faz, então, uma busca na base de especialistas do ICARE por especialistas em <programador C++, interface gráfica> obtidos das fontes 1 ou 4 ou 6.

A Tabela 4 mostra os especialistas em C++ e interface gráfica (e respectivas informações disponíveis sobre eles) que podem ser encontrados nas fontes selecionadas.

Tabela 4 Especialistas em C++ e interface gráfica da SOFTDEV

	Roberto Souza	Luciano Moraes	Alice Cruz
Contato			
E-mail	rs@softdev.com	lm@softdev.com	ac@softdev.com
MSN	---	lm@msn.com	alice@hotmail.com
Ramal	513	794	347
Contexto			
Experiência	3 anos	2 anos	1 ano
Atividade	Escrevendo relatório	Programando	Programando
Cargo	Gerente de projeto	Analista	Programador pleno
Reputação	Ótimo	Muito bom	Muito bom
Disponibili-dade	0.6	0.4	0.8

Considerando que a dificuldade de José requer habilidade prática em programação C++, o ICARE seleciona apenas especialistas que estejam em cargos mais parecidos com o de José e, portanto, executando tarefas mais parecidas. Assim, a lista de especialistas obtida pelo **Recuperador de Especialistas** é a seguinte:

- Luciano Moraes
- Alice Cruz

A lista de especialistas obtida precisa então ser ordenada de forma cômoda para José. Sabe-se pelo contexto dele (Tabela 2) que José está com um prazo curto para o término de sua tarefa e que é um novato na empresa. Portanto, é provável que José precise de um especialista com certa disponibilidade que possa dar uma ajuda mais próxima de uma tutoria, ou seja, alguém que possa ajudar mais de perto. Assim, o ICARE decide ordenar os especialistas encontrados por disponibilidade. Ou seja, prioriza a apresentação daqueles especialistas que podem ajudar José de forma mais ágil.

Mas para isso, o ICARE precisa obter o contexto atual desses especialistas. O CoMGroup dos especialistas busca informações como as descritas na Tabela 5. Considerando o número de atividades em que cada especialista está envolvido e seus prazos, o ICARE calcula o grau de disponibilidade de cada especialista (valor entre 0 e 1). Assim, o ICARE ordena a lista de especialistas encontrados dando prioridade às pessoas menos ocupadas no momento:

1. Alice Cruz
2. Luciano Moraes

A lista ordenada de especialistas é enviada para o módulo **Apresentador de Especialistas**, que gera um documento HTML com a lista dos especialistas e suas informações disponíveis, como nome, contato (e-mail, ramal, MSN) e localização, se possível. Por exemplo, o especialista Luciano está logado na rede interna da SOFTDEV na estação de trabalho 54 e, segundo o mapa da rede, a estação 54 localiza-se na sala 5 do 3º andar. Os resultados obtidos também podem ser apresentados em outros formatos, como XML ou documento de texto.

Tabela 5 Possíveis informações de contexto do especialista

Fonte de contexto	Informação contextual
Google Calendar[6]	Agenda/atividades
MS Project	Cronograma de trabalho
MSN	Status
Currículo	Experiência

A lista de especialistas resultante da busca, formatada para apresentação, é devolvida para o Agente Recomendador. Ele então se encarrega de mostrar para José os especialistas em programação C++ e interface gráfica disponíveis na SOFTDEV no momento. José analisa os especialistas encontrados e decide entrar em contato com Alice porque ela tem boa reputação e encontra-se mais disponível, além de ter um cargo de nível mais próximo ao de José. Por MSN, José explica seu problema e pede a opinião de Alice. Alice se dispõe a olhar o código de José e marca um horário com ele no dia seguinte. Com vinte minutos de conversa com Alice em frente ao código, José sana suas dúvidas e resolve o problema que estava enfrentando.

7. CONCLUSÃO E TRABALHOS FUTUROS

No dia-a-dia de trabalho das pessoas, não é raro perder-se oportunidades de colaboração por simplesmente não se saber que essas oportunidades existem. Essa colaboração não planejada ou estruturada é uma forma informal de colaboração, que pode consistir apenas de uma discussão de idéias ou conceitos e exposição de dúvidas e sugestões. Mas nada impede que uma interação desse tipo incentive o surgimento de um grupo de trabalho organizado com foco em um interesse comum, descoberto informalmente.

Para favorecer o surgimento desse tipo de interação, propomos neste trabalho um sistema que utiliza o contexto de trabalho dos usuários para perceber suas necessidades de colaboração com outras pessoas e recomendar especialistas em um campo de conhecimento útil para o usuário no momento.

[6] http://www.google.com/calendar

O ICARE (*Intelligent Context Awareness for Recommending Experts*) detecta interesses ou problemas do usuário para recomendar uma pessoa que conheça do assunto e que, portanto, seja um potencial colaborador informal. Assim, o ICARE aproxima pessoas buscando informações sobre um determinado assunto com pessoas que provavelmente possuem conhecimento sobre o mesmo.

O ICARE utiliza o contexto de trabalho do usuário para definir que tipo de especialista melhor se adequa às suas necessidades atuais. Também permite o cadastro de diferentes fontes de especialistas, que são selecionadas para uso em cada recomendação de acordo com o contexto do usuário. Por essas características, o ICARE se destaca em relação aos SRE comuns, que são inflexíveis quanto à caracterização dos especialistas e de suas fontes de informações. Além disso, os SRE existentes não levam em conta quem é a pessoa que vai receber a recomendação e como essa recomendação pode ser mais personalizada.

O contexto do especialista também pode ser considerado, possibilitando que o ICARE classifique os resultados encontrados segundo diferentes critérios, como disponibilidade ou proximidade física.

Como trabalhos futuros, pretendemos avaliar a usabilidade do ICARE através de um estudo de caso em um ambiente organizacional real. Além disso, planeja-se analisar e melhorar a cobertura e precisão do Recuperador de Especialistas para retornar o maior número possível de especialistas que atendam com exatidão às necessidades do usuário. Também pretendemos discutir a concordância do especialista em ser recomendado. Afinal, para que a recomendação seja bem sucedida, o especialista deve estar disposto a colaborar. Uma idéia inicial é que o ICARE, uma vez adquirido um e-mail do especialista, entre em contato de maneira automática. Ou seja, o ICARE envia um e-mail para cada possível especialista a ser recomendado explicando o objetivo do sistema e pedindo que ele responda ao e-mail apenas se desejar ser uma pessoa recomendada a outras. Se um especialista aceitar, é levado em consideração pelo Gerador de Base de Especialistas.

REFERÊNCIAS

1. Bazire, M., Brézillon, P. "Understanding Context Before Using It". In: 5th International and Interdisciplinary Conference, CONTEXT 2005, v. LNAI 3554, Springer Verlag, Paris, France (2005), pp. 29-40.
2. Borges, M. R. S., Brézillon, P., Pino, J. A., Pomerol, J.-C. "Bringing context to CSCW". In: Proceedings of the Computer Supported Cooperative Work in Design, v. 2, IEEE Press (2004), pp. 161-166.
3. Chen, H., Finin, T. "An Ontology for Context-Aware Pervasive Computing Environments", The Knowledge Engineering Review, v. 18, n. 3 (2004), pp. 197-207.
4. Dey, A. K. "Understanding and Using Context", Personal and Ubiquitous Computing, v. 5, n. 1 (2001), pp. 4-7.
5. Gutwin, C., Greenberg, S., Blum, R., Dyck, J. "Supporting Informal Collaboration in Shared-Workspace Groupware" (2005).
6. Hinds, J. P., Pfeffer, J. "Why Organizations Don't "Know What They Know": Cognitive and Motivational Factors Affecting the Transfer of Expertise". In MIT Press, Sharing Expertise - Beyond Knowledge Management, chapter 1 (2003).
7. Kautz, R., Selman, B., Shah, M. "ReferralWeb: Combining Social Networks and Collaborative Filtering ", Communications of the ACM, v. 40, n. 3 (1997), pp. 63-65.
8. McArthur, R., Bruza, P. "Discovery of Implicit and Explicit Connections between People Using Email Utterance" (2003), pp. 21-40.
9. McDonald, D. W., Ackerman, M. S. "Expertise Recommender: A Flexible. Recommendation System and Architecture" (2000), pp. 231-240.
10. Reichling, T., Schubert, T., Wulf, V. "Matching Human Actors based on their Texts: Design and Evaluation of an Instance of the ExpertFinding Framework", New York (2005).
11. Santoro, F. M., Brézillon, P., Araújo, R. M. "Context Dynamics in Software Engineering Process", International Journal of Advanced Engineering Informatics, v. Accepted, to appear (2005).
12. Vieira, V. "Gerenciamento de Contexto em Sistemas Colaborativos", Exame de Qualificação e Proposta de Tese de Doutorado, Centro de Informática-UFPE, Brasil (2006).
13. Vieira, V., Tedesco, P., Salgado, A. C. "Towards an Ontology for Context Representation in Groupware". In: Proceedings of the 11th International Workshop on Groupware (CRIWG'05), Porto de Galinhas, Brasil (2005), pp. 367-375.

Percepção de Contextos de Atividade para Identificação de Comunidades Potenciais

Eliane Maria De Bortoli
UTFPR
Avenida Sete de Setembro, 3272
Curitiba, Brasil
+55 46 3223-1805
elianedb@fadep.br

Cesar Augusto Tacla
UTFPR
Avenida Sete de Setembro, 3165
Curitiba, Brasil
+55 41 3310-4685
tacla@cpgei.cefetpr.br

Fabrício Enembreck
PUC-PR
Rua Imaculada Conceição, 1155
Curitiba, Brasil
+55 41 3271-1669
fabricio@ppgia.pucpr.br

RESUMO

Através das redes de computadores é possível que colaboradores de uma organização, física ou geograficamente dispersos, compartilhem um mesmo ambiente de trabalho de forma virtual. Interações ocorrem nesses ambientes porque indivíduos realizam atividades que possuem pontos de intersecção, tais como, relações pessoais e assuntos de interesse. Nesse caso, a percepção das atividades de outros indivíduos se torna essencial, pois o fluxo e naturalidade do trabalho são dificultados em função da distância e da impessoalidade inerentes. A fim de aumentar a percepção dos indivíduos que realizam grande parte de suas atividades em computadores interconectados, esse trabalho visa apresentar um modelo computacional que permite capturar e representar o contexto de atividade dos indivíduos e colocá-los em correlação para identificar comunidades potenciais. O modelo de percepção permite aos indivíduos perceberem outros que realizam ou realizaram atividades em contextos similares. Apresenta-se uma técnica que se distingue das técnicas normalmente empregadas em percepção, pois a maioria delas não leva em consideração o conteúdo pertencente ao contexto de trabalho dos indivíduos, considerando apenas a presença/ausência dos indivíduos ou as atividades realizadas nesse contexto o que dificulta e torna imprecisa a identificação de agrupamentos desses indivíduos.

ABSTRACT

Employees, physically or geographically dispersed, of an organization may share a virtual environment of work. Interactions in real environments result from individuals having common points in their activities or by necessity of coordinating their actions among other reasons. In virtual environments human perception is limited, and it is necessary to introduce it by some mechanism. To perceive the activities of other individuals in virtual environments is important in the flow and naturalness of the work. In order to increase the perception of individuals that perform great part of their activities in networked computers, this work presents a computational model that allows to represent the context of their activities and to put them in correlation to identify potential communities of employees. The perception model allows the individuals to perceive others that perform activities whose contexts are similar. This work distinguishes from others because it takes into consideration the contents of the documents, i.e. the traces of an activity in order to build its context.

Palavras-chave

Percepção. Contexto de atividades. Identificação de comunidades.

Keywords

Perception. Context of activity. Identification of communities.

1. INTRODUÇÃO

A crescente utilização das redes de computadores na realização de atividades nas organizações e, especialmente, o uso da Internet como ferramenta de trabalho tem revelado um ambiente propício à criação de comunidades, englobando tanto o ambiente interno como o externo às organizações. LeFever [16] define estas comunidades virtuais como um grupo de pessoas com interesses comuns que usam a Internet (*sites WEB, e-mail,* programas de mensagens instantâneas) para se comunicar, trabalhar juntos e perseguir seus interesses através do tempo.

A utilização dos computadores em rede permite que colaboradores de uma organização, física ou geograficamente dispersos, compartilhem virtualmente um mesmo ambiente de trabalho. Indivíduos que dificilmente teriam contato, caso não estivessem interconectados por uma rede, podem se beneficiar de interações à distância mediadas por computador. Indivíduos localizados no mesmo ambiente físico, além das interações habituais do ambiente físico, tais como falar ao telefone, cruzar com um colega no corredor, no café, escutar a conversa de outros colegas na mesa ao lado ou "espiar" por cima dos ombros, também utilizam a modalidade virtual de interação.

Há várias pré-condições e motivos para que uma interação ocorra. Como pré-condição tem-se a existência de um canal de comunicação e a afinidade entre os indivíduos. Diversos outros

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 48-57.

motivos podem levar as pessoas a interagirem: trocar informações sobre um assunto completamente alheio ao trabalho (ex. lazer) ou que tenha alguma relação com a atividade da organização (ex. algum assunto técnico). Indivíduos interagem porque realizam ou realizaram atividades que tratam de temas comuns ou envolvem relações pessoais comuns.

Num ambiente virtual a percepção é diminuída, ou seja, não se pode "espiar" por cima dos ombros, nem escutar conversas de corredor, observar se um colega está muito atarefado, estressado ou cansado para, por exemplo, decidir o melhor momento de interrompê-lo. Não se pode nem mesmo saber o que os colegas estão fazendo. Afinal, perceber é adquirir conhecimento, por meio dos sentidos, do que está acontecendo e do que as outras pessoas estão fazendo, mesmo sem se comunicar diretamente com elas [1].

A comunidade de CSCW (*Computer Supported Collaborative Work*) [12] vem estudando formas de aumentar ou melhorar a percepção em sistemas colaborativos com o objetivo de aumentar a usabilidade de tais sistemas. Sistemas colaborativos devem prover elementos de percepção de forma a permitir a coordenação de tarefas cooperativas, principalmente onde a comunicação direta não ocorre. Além disso, devem prover elementos de percepção para que indivíduos possam interpretar eventos, prever possíveis necessidades e transmitir informações de maneira organizada. Perceber as atividades dos outros indivíduos é essencial para o fluxo e naturalidade do trabalho e para diminuir as sensações de impessoalidade e distância, comuns nos ambientes virtuais [8].

Pesquisas recentes relatam a importância de se encontrar meios para facilitar a percepção em ambientes de trabalho locais ou distribuídos enfatizando a representação de contextos de atividades [3]. Dessa forma, são necessários meios para a representação de contextos de atividade facilitando a percepção de aspectos em comum entre os indivíduos e conseqüentemente a criação de canais de comunicação, o que pode ser facilitado pela identificação de comunidades.

O objetivo deste trabalho é apresentar resultados de uma pesquisa em andamento sobre a identificação de comunidades virtuais baseada em contexto de atividades. Apresenta-se um modelo de percepção para formalizar os conceitos, relações e processos envolvidos no problema. Embora o modelo não esteja completamente acabado, realizou-se uma implementação para avaliá-lo preliminarmente. Os experimentos e primeiros resultados são apresentados.

Inicialmente, na seção 2, são apresentados os conceitos de percepção e, na seção 3, as definições associadas a comunidades. Na seção 4 apresentam-se os trabalhos relacionados. A seção 5 descreve o modelo conceitual proposto de percepção e a seção 6, uma implementação parcial do mesmo. A seção 7 descreve os experimentos realizados e os resultados obtidos. A seção 8 contém as considerações finais acerca do trabalho e prosseguimentos da pesquisa.

2. PERCEPÇÃO

As informações de percepção são necessárias para a coordenação das atividades de uma equipe, bem como a cooperação entre seus membros, contribuindo também para o aumento da usabilidade do sistema computacional. A percepção é inerente ao ambiente de trabalho cooperativo. Mas a percepção pode ser útil não apenas quando indivíduos possuem uma meta comum, mas também quando possuem interesses em comum com o propósito de apoiar uns aos outros, aprender e promover seu entendimento através de colaboração eletrônica, empregando práticas comuns, trabalhando com ferramentas comuns, compartilhando crenças e sistemas de valores semelhantes [5].

Quando duas ou mais pessoas colaboram na realização de alguma atividade, uma das principais dificuldades encontradas é a falta de conhecimento do contexto das atividades dos demais indivíduos. Um **contexto** consiste em alguma informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto considerado relevante para os usuários e as aplicações [4].

Usuários que trabalham juntos precisam de informações adequadas sobre o ambiente cooperativo, tais como: presença de outros membros e atividades, compartilhamento de artefatos, entre outros [9]. Informações de **percepção** tornam os indivíduos cientes do estado das entidades presentes no ambiente de trabalho facilitando a interação.

A percepção de contextos de atividades permite que os indivíduos ajam de forma pró-ativa no intuito de colaborar entre si. Em se tratando de comunidades virtuais no âmbito de uma organização, a percepção do contexto de atividades de um grupo de indivíduos possui papel fundamental na identificação de pessoas com interesses em comum ou que possuam conhecimentos ou competências relevantes para os demais indivíduos da organização.

A percepção pode ser classificada dentro de quatro categorias principais [13] *apud* [11]:

- Informal: a percepção informal é o conhecimento de quem está ao redor, o que estas pessoas estão fazendo e o que provavelmente irão fazer. Essas informações podem ser conseguidas a partir do contexto de trabalho de cada indivíduo. A percepção informal é um pré-requisito para a interação espontânea.

- Social: trata da percepção de diferentes tipos de informações subjetivas, tais como: interesse, atenção ou estado emocional de um indivíduo. Isso é freqüentemente percebido através de estímulos não verbais, ou seja, pelo contato visual, expressão facial e linguagem corporal.

- Grupo-estrutural: esse tipo de percepção inclui informações sobre o próprio grupo e seus membros, tais como os papéis e responsabilidades dos membros, o posicionamento e o estado de um membro em relação a determinados assuntos, ou em relação a um artefato compartilhado e os processos do grupo [7].

- Espaço de trabalho: inclui informações sobre o espaço de trabalho em geral, tais como interações de outros participantes no espaço compartilhado e os artefatos nele contidos.

O modelo proposto proporciona uma mescla de percepção informal e do espaço de trabalho conforme ilustrado na Figura 1. A percepção informal se deve ao fato de que os indivíduos terão a possibilidade de acompanhar a atuação dos demais pela representação de seu contexto de trabalho. Já a percepção de espaço de trabalho se configura pela disponibilidade de informações sobre o contexto de trabalho dos indivíduos através de seus artefatos. Estas informações contribuirão para a formação

de comunidades com o intuito de favorecer a colaboração entre indivíduos.

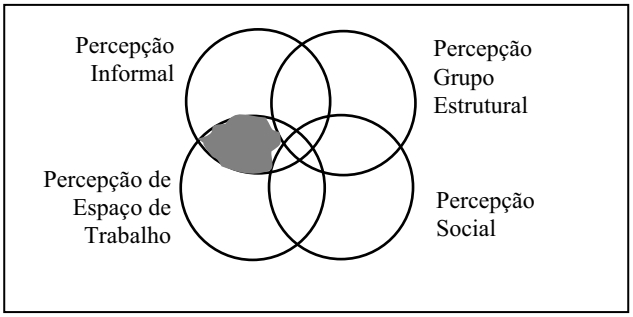


Figura 1. Tipos de percepção aplicados ao modelo proposto.

3. GRUPOS, EQUIPES E COMUNIDADES

Os termos grupos, equipes e comunidades se fazem presentes quando o assunto é colaboração e cooperação entre indivíduos, o que torna importante defini-los adequadamente, diferenciando-os um do outro.

Segundo Schlichter et al. [20], membros de equipes conhecem uns aos outros e colaboram para atingir uma meta comum, enquanto membros de comunidades possuem apenas interesses e preferências comuns. As equipes são normalmente formadas a partir de uma decisão de gerenciamento, selecionando os membros de acordo com seu perfil, competências e potencial de contribuição para um objetivo específico da equipe. Normalmente as equipes são agrupamentos cuja interação é forte, e onde os interesses da equipe prevalecem sobre os interesses pessoais dos membros.

Quanto às comunidades, os autores dizem que essas não possuem uma meta comum e desta maneira, a interação entre os membros da comunidade normalmente é livre. Na maioria dos casos os membros não conhecem uns aos outros e os interesses pessoais prevalecem sobre os interesses da comunidade. Dessa forma, pode-se utilizar o termo grupos para comunidades onde os membros conhecem uns aos outros, mas não necessariamente cooperam. Como exemplos de grupos podem ser citados: grupos de amigos ou membros de um instituto de pesquisa. O termo equipe é considerado como a forma mais avançada de comunidade. A Figura 2 apresenta uma pirâmide representando os três termos.

Muitas são as definições encontradas na literatura para o termo comunidade. De acordo com Mynatt [17], comunidades são consideradas como um "agrupamento social que apresenta diversos níveis: relações compartilhadas no espaço, convenções sociais, um sentido de parceria limitada e um ritmo progressivo de interação social".

Beamish [2] explica que o significado de comunidade giraria em torno de dois sentidos mais comuns. O primeiro refere-se ao lugar físico, geográfico, como a vizinhança, a cidade, o bairro. Dessa forma, indivíduos que habitam em um determinado lugar estabelecem relações entre si devido à proximidade física e vivem sob convenções comuns. O segundo sentido refere-se ao grupo social de qualquer tamanho que divide interesses comuns, sejam religiosos, sociais ou profissionais. Percebe-se então, que Beamish [2] separa o conceito sob dois aspectos: *(i)* o do território como elemento principal na constituição da comunidade e *(ii)* o do interesse comum (e neste caso, o território comum não é mais condição para a existência das relações entre as pessoas) como núcleo da constituição da comunidade.

No âmbito da informática, as comunidades são auxiliadas por sistemas em sua maioria concentrados no processo de construção, ou seja, que facilitam a busca de indivíduos com interesses similares. Nos sistemas de *groupware*, o foco está no processo de colaboração, ou seja, na sincronização e na troca de informações no contexto de uma atividade específica de uma equipe.

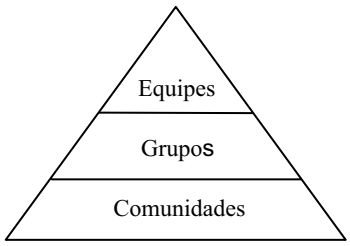


Figura 2. Equipes, Grupos e Comunidades.

Existem diversas ferramentas de software que suportam os processos de construção e de interação de indivíduos em comunidades. De forma geral, são ambientes com indicadores de presença e ferramentas de *chat* integrados a páginas WEB.

As ferramentas de suporte a equipes e a comunidades têm se desenvolvido independentemente apesar de possuírem em comum o propósito de facilitar o contato entre indivíduos que se conhecem ou não. Dessa forma, o modelo de percepção é proposto como base comum para ambos os tipos de sistemas: nas comunidades, para mediar a construção de relações interpessoais e em *groupware,* para aumentar o compartilhamento de informações e eventualmente facilitar a coordenação das ações de uma equipe de trabalho.

4. TRABALHOS RELACIONADOS

Diversos mecanismos são apresentados na literatura para a formação de comunidades e que se baseiam em percepção, assim como Cyclades [10], Piazza[14], entre outros. Nesta seção serão apresentados brevemente alguns de seus pontos mais importantes. Mediante análise desses mecanismos, é possível classificar as ferramentas existentes para esse fim em três categorias em função da forma de representação do contexto:

- análise de conteúdo: ferramentas que se baseiam na verificação do conteúdo manipulado pelos usuários;

- análise de localização: ferramentas que se baseiam na estrutura de *hiperlinks* referenciados pelos usuários;

- via interface de conexão: ferramentas onde a formação de comunidades ocorre pela associação voluntária dos usuários através de um sistema *online*.

Dentre as ferramentas que objetivam a formação de equipes ou comunidades baseando-se no conteúdo dos elementos (ex. documentos, páginas WEB) pertencentes ao contexto de trabalho dos usuários, tem-se que a forma como são exploradas as

informações diferem em alguns aspectos. Jena [6] utiliza-se de ontologias e é direcionado para o gerenciamento de um projeto de pesquisa científica, o que permite perceber o significado do conteúdo gerado pelos membros da comunidade.

O Cyclades [10] disponibiliza um ambiente de arquivos composto por um grande número de arquivos multidisciplinares e heterogêneos. O espaço de percepção explora ligações explícitas ou tácitas entre membros de organizações que atuam em atividades semelhantes, através de informações e contatos pessoais que de outra maneira poderiam perder-se. Dessa maneira, essas informações e contatos pessoais são captados e armazenados em bases de dados a fim de tornar os usuários cientes de pessoas que tenham procurado a mesma informação ou escrito documentos similares.

Kamahara et al. [15] propuseram um sistema baseado em filtragem colaborativa – sobre o perfil de usuários - agregada à filtragem baseada em conteúdo – que é aplicada sobre o perfil dos demais usuários. Tal sistema, direcionado à recomendação de programas de televisão, envolve três elementos: os usuários, os itens (conteúdos multimídia) e as comunidades (grupos de usuários). Cada elemento possui seus próprios atributos. Os atributos dos elementos são chamados perfis. Um perfil de usuário contém suas preferências e é representado por um vetor de palavras-chave que é usado no cálculo das similaridades. Sendo assim, os usuários que possuem maior similaridade recebem recomendações de programas de televisão de seu interesse.

Piazza [14] é uma ferramenta que permite aos indivíduos estarem cientes sobre quem trabalha em tarefas similares facilitando interações involuntárias. Além disso, suporta contatos intencionais e reuniões de planejamento.

Semelhante ao Piazza [14], Vivacqua et al. [22] propõe o CUMBIA, um sistema que faz uso de agentes para detectar oportunidades para colaboração de forma dinâmica. Trata-se de um *framework* baseado em agentes onde cada usuário possui um grupo de agentes para auxiliá-lo na gestão do conhecimento e tarefas colaborativas. Agentes identificam possíveis situações de cooperação extraindo informações do contexto de trabalho e do perfil dos usuários. Mais especificamente, essas informações envolvem: as tarefas recentes, documentos criados ou acessados recentemente, histórico de acesso a páginas WEB (tomados a partir do *cache* de acesso à Internet e histórico) e atividades de cada usuário. Os agentes coletam tais informações continuamente e as enviam aos agentes de outros usuários. Então, é realizada uma comparação – cálculo da proximidade de um usuário aos demais - a fim de encontrar similaridades entre seus contextos de trabalho. Dessa forma, indivíduos que executam tarefas similares podem ser contatados por outros.

O sistema I2I [3] também se baseia na similaridade de interesses dos usuários para agrupá-los, mas nesse caso o que os une é o fato de visitarem páginas WEB com conteúdos similares. O conteúdo de uma página WEB é representado pelos seus termos mais representativos.

Considerando os trabalhos apresentados, é possível definir duas formas de extrair informação para construir contextos de atividades (que irão determinar perfis de usuários) e, em seguida, agrupar usuários em comunidades, são elas: *(i)* as que extraem informações do conteúdo dos recursos existentes no contexto e *(ii)* as que exploram a estrutura de conexão dos recursos, i.e., os *links* entre recursos ou usuários.

Nos trabalhos que se preocupam com o conteúdo dos elementos manipulados pelos usuários em suas sessões de trabalho a maior dificuldade está na captura de informações utilizadas para representar o contexto de uma atividade. Como principal limitação pode-se citar a existência de dados confidenciais dos indivíduos e da própria organização, os quais normalmente não são disponibilizados. Essas limitações podem levar a possíveis inconsistências ao se efetuar o agrupamento dos indivíduos em comunidades por falta de informação sobre suas atividades reais.

O modelo de percepção proposto neste trabalho se baseia na análise do conteúdo dos documentos eletrônicos manipulados pelos indivíduos. A abordagem apresenta fortes semelhanças com CUMBIA [22] e I2I [3], se distinguindo pelo fato de propor uma análise temporal dos contextos. Isto possibilita traçar a evolução das comunidades em função de seus interesses ao longo do tempo, ou seja, detectar os temas que deixam de ser relevantes e os que passam a serem considerados como tais pelos indivíduos.

5. MODELO CONCEITUAL PROPOSTO

O modelo de percepção está fundamentado no conteúdo dos recursos manipulados pelos usuários. A análise destes conteúdos permite a identificação de comunidades **potenciais** de indivíduos. Os recursos principais são os documentos eletrônicos textuais (ex. arquivos pdf, html) denominados artefatos textuais. Assume-se que os artefatos textuais são vestígios das atividades executadas pelos indivíduos e, portanto, podem ser utilizados para reconstruir o contexto das mesmas. O objetivo é encontrar similaridades entre os contextos que permitam o agrupamento desses usuários em comunidades afins.

Conceitualmente propõe-se um modelo de percepção que permite aos indivíduos perceberem outros indivíduos que realizam ou realizaram atividades em contextos similares como ilustra a Figura 3.

Uma atividade é definida como um conjunto de ações (ex. abrir um documento, pesquisar na WEB, conversar com um colega) realizadas por um indivíduo visando atingir um objetivo (ex. obter informação, aprender algo, construir um artefato). Dessa forma, atividade assume aqui um sentido genérico, tais como atividades profissionais, de lazer, de informação e outras. Um indivíduo realiza diferentes atividades ao longo do tempo, cada uma delas em contextos específicos que podem ser reconstruídos a partir dos artefatos textuais acessados ou produzidos.

A mesma atividade realizada em momentos diferentes pode apresentar contextos diferentes. Por exemplo, a atividade "manter-se informado sobre política" pode ser realizada em contextos diferentes. Em um dia, pode-se acessar as páginas de política do jornal A porque houve uma notícia sobre um escândalo de corrupção na câmara dos deputados e, num outro dia, as páginas de política do jornal B para manter-se atualizado sobre as atividades dos candidatos à presidência da república. Assim, o contexto de uma atividade contém informações que descrevem o cenário no período de execução da atividade. Neste cenário, encontram-se:

- Data, horário, local geográfico e local físico [9].

- Os artefatos manipulados, tais como: documentos, desenhos, imagens e outros produtos resultantes da utilização de aplicativos [9].

- As ferramentas utilizadas para manipular os artefatos, tais como, softwares aplicativos.

- As pessoas envolvidas na atividade, seus estados emocionais, a sub-rede social destas pessoas com as relações interpessoais, informações de reputação e confiança.

- Fatos e eventos que influenciam a execução da atividade: durante a atividade de redação de um relatório técnico, como por exemplo, um *e-mail* do gerente chega adiando a entrega do mesmo, assim a atividade pode temporariamente ser suspensa [9].

- Seqüência de ações executadas (operações informáticas, tais como consultar páginas WEB, abrir, criar ou editar um documento texto).

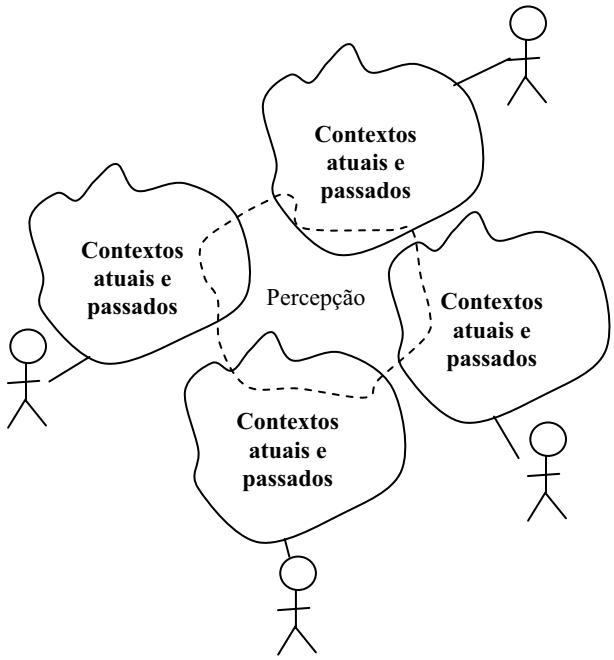


Figura 3. Contexto como conceito fundamental da percepção.

Do ponto de vista de um indivíduo, diversos contextos de atividades podem coexistir em função das atividades em execução ou já executadas. Dois indivíduos quaisquer que realizam o mesmo tipo de atividade dificilmente terão contextos exatamente iguais. Portanto, a percepção de contextos deve buscar contextos similares (não exatamente iguais) e deve ser assíncrona (as atividades não precisam ocorrer ao mesmo tempo).

A análise temporal dos contextos que permite o estudo da evolução (alterações) das comunidades ao longo do tempo, seria realizada através da coleta dos artefatos textuais dos indivíduos durante a realização de suas atividades. Essa análise possui diversas aplicações nas organizações:

- **Descoberta de lideranças**: uma comunidade que permanece estável em relação aos seus membros apresenta, muito provavelmente, um membro condutor/líder. O estudo das relações interpessoais em uma comunidade pode ser aprofundado pelo emprego de ferramentas de análise de redes sociais. A identificação da comunidade é apenas o passo inicial.

- **Descoberta de competências**: comunidades podem revelar competências existentes até então desconhecidas na organização. Os indivíduos podem desenvolver atividades paralelas relacionadas a temas que podem ser de interesse da organização. Cabe a esta propiciar meios para a comunidade se desenvolver.

- **Mapeamento do capital intelectual:** relacionado ao item anterior, a identificação de comunidades transversais aos departamentos de uma organização pode mostrar as áreas de conhecimento da mesma e a quantidade de pessoas envolvidas/interessadas por área. Uma análise mais detalhada pode determinar as áreas que necessitam/merecem investimento e as pessoas chave para desenvolver competências que serão necessárias à organização.

O modelo conceitual de percepção proposto visa identificar comunidades potenciais existentes a partir de uma população de indivíduos que executam atividades não modeladas *à priori*. A identificação de uma atividade é feita pelo contexto da mesma. A identificação dos membros de uma comunidade ocorre em função do cálculo da similaridade dos contextos das suas atividades. Em função do grau de similaridade, um indivíduo pode ser identificado como participante de nenhuma ou de diversas comunidades. A análise temporal das comunidades pode revelar informações importantes sobre o capital intelectual da organização apontando pessoas-chave para o seu desenvolvimento e direções para investir em comunidades que atuam áreas de conhecimento importantes para a organização.

6. IMPLEMENTAÇÃO DO MODELO

Embora o modelo conceitual não esteja completamente maduro, decidiu-se pela realização de uma implementação parcial para avaliar a qualidade das comunidades identificadas utilizando um algoritmo construído pelo nosso grupo de pesquisa (Tacla e Enembreck [21]).

No estágio atual, a implementação não contempla a representação do contexto de atividade na sua totalidade. O contexto é composto pelos dados básicos (data, hora, local) e pelos artefatos textuais, mais precisamente, por uma representação do conteúdo dos artefatos textuais (*at*) – documentos, páginas da WEB e e-mails. As subseções seguintes detalham a construção do contexto de atividade e o cálculo de similaridade entre contextos.

6.1 Construção de um contexto de atividade

Os contextos de atividades (*CAs)* são representados como vetores de termos relevantes. Uma medida comum de relevância para termos é a TF-IDF (*Term Frequency - Inverse Document Frequency*) [18]. TF-IDF especifica que a relevância de um termo em um artefato textual (*at*) está em proporção direta para sua freqüência no *at* e em proporção inversa à sua incidência em toda a coleção de *ats*, representada por *AT*.

O elemento IDF para o i-ésimo termo é dado por $log(|AT|/DF_i)$, onde DF_i é a quantidade de *ats* contendo o termo i. TF_i designa a freqüência do i-ésimo termo em um *at* particular. A fórmula TF-IDF é dada pela equação 1.

$$TFIDF(i) = TF_i \times \log\left({|AT|}/{DF_i} \right) \qquad (1)$$

Um *at* é representado por um vetor conforme mostra a equação 2.

$$\boldsymbol{at}=\{TF_1 * log(AT/DF_1), ..., TF_m * log(AT/DF_m)\} \qquad (2)$$

Um vetor médio representa o contexto de atividade do usuário. A equação 3 retorna um vetor c (centróide) para uma coleção AT de *ats* pertencente a um certo usuário.

$$c = \frac{1}{|AT|} \times \sum_{at \in AT} at \qquad (3)$$

Cada vez que um *at* é criado, acessado ou modificado durante uma atividade, torna-se necessário atualizar os vetores de representação dos *ats*. Cada vez que um contexto é alterado (o vetor médio dado pela equação 3), serão alterados os valores de similaridade entre contextos.

6.2 Cálculo da Similaridade

Os contextos de atividade de cada usuário possuem diferentes termos e conseqüentemente diferentes dimensões. Naturalmente, os contextos de atividade podem possuir termos comuns, dependendo da similaridade dos conteúdos dos seus *ats*. Mas para realizar o cálculo da similaridade, primeiramente é preciso normalizar os contextos de atividade para comparar e descobrir os termos que melhor discriminam esses contextos. Um termo que é comum, ou seja, é importante para muitos usuários (ex. "projeto") não seria um bom discriminador.

6.2.1 Poder de discriminação dos termos

Para medir o poder de discriminação dos termos encontrados nos contextos de atividade é utilizada a técnica de índice Gini [19]. Considera-se $\{c_1, c_2, ..., c_m\}$ como sendo um conjunto de contextos de atividade calculados de acordo com a equação (3) e T_i, o vetor derivado a partir da relevância do termo i em todos os contextos tal que $T_i = \{c_{1i}, c_{2i}, \ldots, c_{mi}\}$. T'_i é o vetor normalizado, tal que $T'_i = \{c_{1i} / \| T_i \|_1, c_{2i} / \| T_i \|_1, c_{mi} / \| T_i \|_1\}$ e $\| T_i \|_1$ é a norma unitária do vetor T_i (somatório do módulo de todos os elementos do vetor T_i). O poder de discriminação do termo *i* – denominado p_i – é dado pela equação 4.

$$p_i = \sum_{j=1}^{m} T'^2_{ji} \qquad (4)$$

Para cada termo *i*, p_i é igual ao somatório dos quadrados dos elementos do vetor T'_i. O valor de p_i está sempre no intervalo [1/m, 1]. p_i apresenta valor mais baixo quando $T'_{1i} = T'_{2i} = \ldots = T'_{mi}$, ou seja, quando o termo possui a mesma relevância em todos os contextos. O valor mais alto de p_i ocorre quando apenas um contexto de atividade possui o termo *i*.

6.2.2 Similaridade

Para quantificar a similaridade entre dois contextos de atividade c_1 e c_2, é criado um vetor comparável c'_2, da seguinte forma. Para cada termo c_{1i}, o correspondente c'_{2i} é comparado com o c_{2i}. Quando um termo c_{1i} existe em c_2 então c'_{2i} é o resultado de *mínimo*(c_{1i}, c_{2i}), caso contrário atribui-se zero à c'_{2i}. Termos existentes somente em c_2 não são copiados parra c'_2.

A similaridade entre c_1 e c_2 é computada utilizando-se do poder de discriminação dos termos (*pi*), de acordo com a equação 5. Assume-se que o valor máximo de similaridade é alcançado quando se compara um vetor c_i com ele mesmo. Daí a utilização de *mínimo*(c_{1i}, c_{2i}) na composição de c'_2 no parágrafo anterior.

$$similaridade(c_1, c'_2, p) = \frac{\sum_{i=1}^{|c_1|} c_{1i} \times c'_{2i} \times p_i}{|c_1|} \qquad (5)$$

O cálculo de similaridade pode ser visto como um algoritmo de classificação capaz de manipular diversas classes não conhecidas à priori (algoritmo não supervisionado). A justificativa para isso é que a aplicação colaborativa pretendida consiste de diversos usuários, cada usuário representando uma classe (contexto de atividade) no sistema. Os usuários com contextos similares são os exemplares a serem classificados. Em tal aplicação a técnica de classificação deve ser flexível para suportar inserção e remoção de usuários.

6.3 Exemplo

A Tabela 1 ilustra a representação dos contextos de atividades para três usuários (c_1, c_2 e c_3). Assim, o usuário 1 possui um contexto formado pelos termos T_1, T_2 e T_3. Cada posição dessa tabela contém o valor médio do TF/IDF por termo e por usuário. Estes valores são calculados com base na equação 3.

Tabela 1. Contextos das atividades dos usuários 1, 2 e 3.

Termo	c_1	c_2	c_3
T_1	0,6361		
T_2	3,2283		
T_3	0,4771	0,9542	
T_4		0,2347	0,4771
T_5		0,6361	0,2347
T_6			0,3180
T_7			0,6361

Tabela 2. Vetores normalizados T'_i e índice Gini.

Termo	c_1	c_2	c_3	p_i
T'_1	1,0000			1,0000
T'_2	1,0000			1,0000
T'_3	0,3333	0,6667		0,5555
T'_4		0,3298	0,6702	0,5579
T'_5		0,7304	0,2696	0,6061
T'_6			1,0000	1,0000
T'_7			1,0000	1,0000

Em seguida, calcula-se o índice Gini (última coluna da Tabela 2) para cada um dos termos de acordo com a equação 4. Os vetores T'_i, necessários para o cálculo de p_i, também estão ilustrados na Tabela 2 (os elementos da coluna p_i não fazem parte dos vetores T'_i). Nota-se que os termos T_1, T_2, T_6 e T_7 são os melhores discriminadores, pois só aparecem em um dos contextos.

Para calcular a similaridade entre contextos, é preciso construir vetores comparáveis de mesma dimensão, como indicado na seção 6.2.2. A Tabela 3 mostra estes vetores tomando-se como vetor base o contexto do usuário 1 (c_1).

Tabela 3. Vetores para comparação com o contexto c_1.

Termos	C'_1	c'_2	c'_3
T_1	0,6361	0	0
T_2	3,2283	0	0
T_3	0,4771	0,4771	0

Finalmente, pode-se comparar o vetor c_1 com os vetores c'_i. A Tabela 4 mostra os resultados da aplicação da equação 5. A maior similaridade é obtida quando se compara o vetor c_1 com ele mesmo. Em segundo lugar, c_1 com c_2 e, finalmente, c_3, que não tem nenhum termo comum com c_1, apresenta similaridade zero.

Tabela 4. Cálculo da similaridade.

Termos	c_1 x c'_1	c_1 x c'_2	c_1 x c'_3
T_1	0,40470	0	0
T_2	10,4221	0	0
T_3	0,12646	0,126469	0
Similaridade	**10,9533**	**0,126469**	**0**

7. EXPERIMENTAÇÕES E RESULTADOS

A implementação do modelo contempla a parte de construção dos contextos de atividades e do cálculo de similaridade. A coleta dos artefatos textuais foi realizada manualmente assim como a coordenação dos processos computacionais de construção dos contextos e de cálculo de similaridade.

7.1 Metodologia

Utilizou-se uma experimentação controlada onde a população amostrada era conhecida, isto é, os grupos de pertinência dos indivíduos eram previamente conhecidos. Foram selecionados cinco usuários (A, B, C, D e E) que participam de dois grupos distintos de pesquisa: aplicações de sistemas multi-agentes e otimização em planejamento e organização. Os usuários são assim classificados:

- **usuários A e B:** pertencem ao grupo de sistemas multi-agentes. O usuário A trabalha em CSCW e o usuário B, em problemas de empacotamento e armazenagem. Até o momento da coleta dos artefatos textuais, os usuários A e B não haviam trabalhado especificamente na matéria agentes.

- **usuários C, D e E:** pertencem ao grupo de otimização. Todos trabalham em otimização de planejamento e organização aplicada a plantas industriais.

Cada usuário selecionou aproximadamente 10 artefatos textuais (*ats*) produzidos ou acessados durante suas atividades de pesquisa durante 3 (três) meses. Um *at* típico é um artigo científico contendo de 6 (seis) a 20 (vinte) páginas na língua inglesa. Realizaram-se cinco experimentos para avaliar a qualidade dos resultados produzidos pelo algoritmo apresentado na seção 6 na identificação dos grupos. Em segundo lugar, desejava-se estudar a influência da quantidade de termos que representam um contexto de atividade na identificação dos grupos. Quanto menos termos, menos recursos computacionais são necessários para construção dos contextos e cálculo das similaridades. Assim, no primeiro experimento, foram utilizados 200 termos para representar o contexto de cada usuário. Nos subseqüentes, foram utilizados 500, 1.000 e 5.000. Para cada contexto de atividade calculou-se a similaridade com todos os outros produzindo os resultados mostrados em seguida.

7.1.1 Resultados do Cálculo da Similaridade

A Tabela 5 mostra os valores obtidos a partir do cálculo da similaridade quando utilizado um vetor de tamanho 200 representando os contextos de atividade. A interpretação da tabela é feita linha a linha. Assim, para o usuário C, o contexto de atividade com maior similaridade (exceto para seu próprio contexto) é D, e o segundo mais similar é E. Os valores de similaridade são dados pela equação 5. A Tabela 6 mostra os valores de similaridade para contextos representados com 5.000 termos.

Tabela 5. Valores de similaridade para contextos de atividade com 200 termos. Os contextos mais similares estão em negrito, e os segundos colocados estão em itálico.

Usuário	A	B	C	D	E
A	2.41206	0.07356	**0.23636**	0.07605	***0.14382***
B	0.06790	2.13679	***0.11372***	0.07659	**0.13650**
C	0.09142	0.04534	5.34660	**0.48570**	***0.44361***
D	0.02228	0.05716	**0.65757**	3.33189	***0.55121***
E	0.02955	0.03785	**0.43324**	***0.26120***	3.62506

Tabela 6. Valores de similaridade para contextos de atividade com 5.000 termos. Os contextos mais similares estão em negrito e os segundos colocados estão em itálico.

Usuário	A	B	C	D	E
A	1.00000	0.08779	**0.17062**	***0.10591***	0.06488
B	***0.09809***	0.62086	**0.17059**	0.09780	0.07915
C	0.08996	0.07276	0.93100	**0.20679**	***0.12578***
D	0.04809	0.04107	**0.19441**	0.75614	***0.09824***
E	0.05333	0.07633	**0.21551**	***0.15317***	0.83853

Observa-se que quando o tamanho dos vetores é aumentado (Tabela 6), ocorreram poucas modificações na posição dos primeiros e segundos colocados em relação à tabela 5:

- Na Tabela 6, linha A, o contexto D assumiu a segunda colocação, antes ocupada por E (tabela 5);

- Na Tabela 6, linha B, o contexto C assumiu a primeira posição no lugar de E. O contexto A assumiu a posição de segundo lugar antes ocupada por C.

- Nas demais linhas não ocorreram modificações.

Quando utilizado um vetor de tamanho 1.000, ocorreu apenas uma modificação e com vetores de tamanho 500, nenhuma. Isto pode indicar pouca influência do tamanho do vetor no cálculo da similaridade, mas o tamanho reduzido da amostra não permite generalizar o resultado obtido. Contudo foi notado que os valores de similaridade diminuíram à medida que os tamanhos dos vetores aumentaram. Quanto maior o número de termos utilizados, menor a probabilidade de serem bons discriminadores, menor o índice Gini e, conseqüentemente, os valores de similaridade.

7.1.2 Identificação de comunidades

Analisando as medidas de similaridade, foi possível identificar os grupos (caso específico de comunidade) de usuários contendo contextos similares. Se for tomada a tabela 5 como exemplo, nota-se que os usuários C e E são mais similares ao usuário A, mas nem C e nem E tem A entre os dois primeiros mais similares. Isso significa que embora C e E possam colaborar com A, eles não formam uma comunidade porque o usuário A não trabalha em um assunto relevante para C e E.

Considerando a medida de similaridade dos usuários C, D e E, pode-se concluir que eles formam uma comunidade. Pode-se notar que os usuários D e E são mais próximos a C, usuários C e E aproximam-se de D, e os usuários C e D aproximam-se de E. Assim, quando a medida de similaridade é recíproca para um conjunto de usuários pode-se dizer que tal conjunto forma uma comunidade.

De fato, essa análise permite dizer que o algoritmo proposto foi eficiente neste experimento, pois identificou a comunidade de otimização a partir de seus *ats*. Não foi possível identificar a equipe de sistemas multi-agentes porque ambos os usuários haviam trabalhado em assuntos diferentes.

Outro resultado interessante é que é possível determinar o grau de coesão de uma comunidade identificada. A figura 4 apresenta no eixo x os contextos de atividade dos usuários e a distância entre eles sobre o eixo Y. A distância é dada pela relação *similaridade*(c*i*, c*i*, p*i*)/*similaridade*(c*i*, c*j*, p*i*) obtida na tabela 5, resultando na tabela 7. O valor mínimo é 1 quando *i=j*, i.e., quando um contexto é comparado com ele próprio (o contexto mais similar a um contexto é o próprio contexto). O valor máximo para a distância tende a infinito quando o denominador da equação se aproxima de zero.

O gráfico da Figura 4 é a representação gráfica da tabela 7, eliminando-se as distâncias acima de 35 para destacar as distâncias entre os usuários. A e B não são identificados como uma comunidade porque eles são os mais distantes de cada um dos outros (B é a maior distância a partir de A e vice-versa). Considerando o contexto de atividade do usuário C, nota-se que os contextos de D e E são muito próximos, enquanto os contextos de A e B não aparecem na Figura 4, pois apresentam distâncias respectivas de 58,48 e 117,92. O mesmo acontece para os usuários D e E. Considerando que os contextos dos usuários C, D e são muito próximos, pode-se dizer que este grupo possui um alto grau de coesão.

Tabela 7. Distância entre os usuários.

Usuário	A	B	C	D	E
A	1,00	32,79	10,21	31,72	16,77
B	31,47	1,00	18,79	27,90	15,65
C	58,48	117,92	1,00	**11,01**	**12,05**
D	149,55	58,29	**5,07**	1,00	**6,04**
E	122,68	95,77	**8,37**	**13,88**	**1,00**

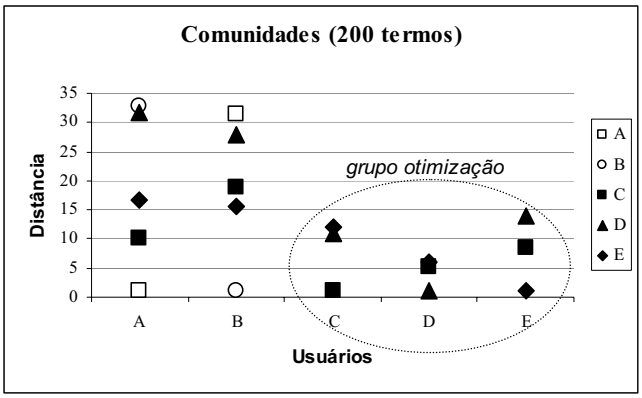


Figura 4. Distância entre os contextos de atividade

Os experimentos realizados sinalizam que o rumo tomado pode dar bons frutos. Obviamente, experimentos com populações maiores e mais heterogêneas precisam ser realizados para se chegar a conclusões mais definitivas.

8. EXPERIMENTAÇÕES E RESULTADOS

Esse trabalho tratou de aspectos relacionados à identificação de comunidades através da análise do conteúdo dos artefatos textuais manipulados pelos usuários que fazem uso de computadores interconectados por uma rede de comunicação. Dessa forma foi proposto um modelo conceitual que pode ser aplicado como um mecanismo de percepção para participantes de grupos acerca das atividades dos seus colegas e, também, na identificação de comunidades. Um dos elementos que distingue o modelo conceitual apresentado de trabalhos similares é a análise da evolução das comunidades ao longo do tempo, embora este assunto necessite maior aprofundamento.

Além do modelo conceitual de percepção proposto, o presente trabalho apresentou uma implementação parcial do mesmo, enfatizando uma técnica baseada em processamento estatístico de textos para construir e comparar contextos de atividades. Os

resultados iniciais ainda não permitem generalizar os resultados obtidos, apenas sinalizam que a direção tomada está correta.

Os trabalhos futuros incluem experimentos mais aprofundados utilizando-se uma população maior e mais heterogênea. A coleta de dados (em andamento) se fará sobre um horizonte de tempo maior (6 meses) para que se possa avaliar a evolução das comunidades. Também se encontra em desenvolvimento um protótipo do sistema que funcione de maneira distribuída em uma arquitetura *peer-to-peer*. O protótipo inclui a definição de um protocolo de notificação aos indivíduos, daqueles que apresentam contextos similares. A maior dificuldade está na coleta automática dos artefatos textuais utilizados pelos usuários nas suas atividades cotidianas devido a questões de direitos de acesso, privacidade e proteção da informação. Um passo posterior será o estudo de uma forma gráfica de visualização das comunidades potenciais.

Em relação aos benefícios da aplicação da abordagem proposta, espera-se:

- agilizar o fluxo de informações interna ou externamente às organizações, contribuindo para o aumento do compartilhamento de informações entre indivíduos e por conseqüência o nível de conhecimento organizacional;

- agilizar os processos de trabalho individuais ou coletivos, através do compartilhamento de interesses;

- identificar competências inerentes a uma comunidade pertencente a uma organização - é preciso muitas vezes agregar competências de um grupo de pessoas ou mesmo descobrir competências que até então não se tinha ciência de sua existência;

- favorecer a criação de um ambiente de aprendizado, havendo o desenvolvimento de novas habilidades profissionais, o que pode ser conseguido, por exemplo, através da transferência de melhores práticas de trabalho entre os indivíduos.

9. REFERENCIAS

[1] Brinck, T. and McDaniel, S. E. *Awareness* in Colaborative Systems, *Workshop Report*, SIGCHI Bulletin, 1999.

[2] Beamish, A.Communities On-line: A Study of Community – Based Computer Networks. Tese de Mestrado em Planejamento de Cidades. Instituto de Tecnologia de Massachusetts – Estados Unidos. 1995. Disponível em http://albertimit.edu/arch/4.207/anneb/thesis/toc.html

[3] Budzik, J., Bradshaw, S., Fu, X. and Hammond, K. J., Clustering for Opportunistic Communication. *WWW 2002*, Honolulu, Hawaii, USA. ACM, 2002.

[4] Dey, A..K. and Abowd, G..D. Towards a Better Understanding of Context and Context-Awareness. Proceedings of the CH12000 Workshop on The What, Who, Where, When, Why and How of Context-Awareness, April, 2000.

[5] Dias, M.S., Xeréo, G. Editor Cooperativo para Diagramação de Software OO. In: XI Simpósio Brasileiro de Engenharia de Software. *Anais*. Fortaleza, CE, outubro, 1997. P. 499-502.

[6] Dias, A.; Welfer, D. e D´Ornellas, M. C. JENA: uma ferramenta para desenvolver comunidades virtuais de pesquisa científica. Revista do CCEI, Vol. 8, Numero 14, Agosto, 2004.

[7] Ellis, C. A.; Gibbs, S. J. e Rein, G. L. Groupware: somes issues and experiences. *Communications of the ACM*, New York, v. 34, n. 1, p. 38.

[8] Fuks, H. and Assis, R. L. Facilitating Perception on Virtual Learningware-Based Environments, *Journal on Systems and Information Technology*, 2001.

[9] Gross, T. and Prinz W. Modelling Shared Contexts in Cooperative Environments: Concept, Implementation, and Evaluation. Computer Supported Cooperative Work 13: 283–303, 2004.

[10] Gross, T., CYCLADES: A Distributed System for Virtual Community Support Based on Open Archives, *Eleventh Euromicro Conference on Parallel, Distributed, and Network-Based Processing - PDP 2003* (Feb. 5-7, Genova, Italy). Clematis, A., ed. IEEE Computer Society Press, Los Alamitos, CA, 2003. pp. 484-491.

[11] Gross, T., Stary, C. and Totter, A. User-Centered Awareness in Computer-Supported Cooperative Work-Systems: Structured Embedding of Findings from Social Sciences. International Journal of Human-Computer Interaction, pp. 323-360, 2005.

[12] Gutwin, C. and Greenberg, S. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work: The Journal of Collaborative Computing 11*, 3-4 (2002). pp. 411-446.

[13] Gutwin, C., Greenberg, S. and Roseman, M. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In *Proceedings of the Conference on Human-Computer Interaction: People and Computers - HCI'96* (Aug. 20-23, London, UK). Springer-Verlag, Heidelberg, 1996. pp. 281-298.

[14] Isaacs, E. A., Tang, J.C. and Morris, T. PIAZZA: A desktop Environment Supporting Impromtu and Planned Interactions. Proc. CSCW'96, Cambridge, MA, 1996.

[15] Kamahara, J., Asakawa, T., Shimojo, S. and Miyahara, H., A Community-based Recommendation System to Reveal Unexpected Interests. Proceedings of the 11th International Multimedia Modelling Conference (MMM'05), IEEE, 2005.

[16] Lefever, L. ***E-mail*** Lists and Message Boards – Where is the Middle Ground? 2003. Disponível em <http://www.commoncraft.com/archives/000401.html>. Acessado em 15/09/2006.

[17] Mynatt, E.; O'Day, V.; Adler, A. and Ito, M., Design for Network Communities. In Proc. ACM SIGCHI Conf. on Human Factors in Compt. Syst., 1997.

[18] Salton, G., Automatic Text Processing: The Transformations, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.

[19] Shankar, S., and Karypis, G., A Feature Weight Adjustment Algorithm for Document Categorization, *KDD-2000 Workshop on Text Mining*, Boston, USA, August 2000.

[20] Schlichter, J.; Koch, M. and Xu, C., Awareness – The Common Link Between Groupware and Community Support

Systems. In T. Ishida, editor, Community Computing and Support Systems, pages pp. 77-93. Springer Verlag, 1998.

[21] Tacla, C. A and Enembreck, F. An Awareness Mechanism for Enhancing Cooperation in Design Teams. The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings. 2005. pp. 920-925.

[22] Vivacqua, A., Moreno, M. and Souza, J. CUMBIA: An Agent Framework to Detect Opportunities for Collaboration. Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design. 2005. pp. 417-422.

Modelo 3C de Colaboração para o desenvolvimento de Sistemas Colaborativos

Mariano Pimentel[1], Marco Aurélio Gerosa[1], Denise Filippo[1],
Alberto Raposo[2], Hugo Fuks[1], Carlos José Pereira de Lucena[1]
Depto. de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) – (21) 31141500
R. Marquês de São Vicente, 225 – 22.453-900 Rio de Janeiro - RJ
[1] {mariano,gerosa,denise,hugo,lucena}@inf.puc-rio.br
[2] abraposo@tecgraf.puc-rio.br

RESUMO

Este artigo apresenta como o Modelo 3C de Colaboração (Comunicação, Coordenação e Cooperação) tem sido apropriado por nosso grupo de pesquisa para o desenvolvimento de sistemas colaborativos. Constata-se que este modelo é útil em diferentes etapas do processo de desenvolvimento: para auxiliar a análise do groupware a ser desenvolvido; no estabelecimento do foco para o desenvolvimento de sucessivas versões do groupware num processo iterativo e investigativo; e para o desenvolvimento de arquitetura e componentes 3C.

Palavras Chaves

Sistemas Colaborativos, Desenvolvimento de Groupware, Modelo 3C de Colaboração

1. INTRODUÇÃO

A colaboração envolve comunicação, coordenação e cooperação. Comunicação se realiza através da troca de mensagens; coordenação se realiza através do gerenciamento de pessoas, atividades e recursos; e cooperação se realiza através de operações num espaço compartilhado para a execução das tarefas. Este modelo, que nosso grupo de pesquisa denomina Modelo 3C de Colaboração, foi originalmente proposto no artigo de Ellis *et al.* [5] com algumas diferenças de nomenclatura.

O Modelo 3C de Colaboração é frequentemente usado pela literatura para classificar os sistemas colaborativos, tal como proposto por Teufel et al. [22] e Borghoff & Schlichter [1]. Algumas tentativas têm sido feitas para usar este modelo no desenvolvimento de groupware. Um exemplo é o modelo de projeto Clover, que define 3 classes de funcionalidades, denominadas comunicação, coordenação e produção [15]. O modelo Clover compartilha da mesma utilidade do Modelo 3C de Colaboração – ambos lidam com as três classes de funcionalidades que uma aplicação colaborativa deve dar suporte. Um modelo similar ao Modelo 3C de Colaboração também já foi usado para guiar a detecção de problemas de usabilidade [17], levando os avaliadores a focarem a atenção nos aspectos de comunicação, coordenação e cooperação.

Nosso grupo de pesquisa investiga o uso sistemático do Modelo 3C de Colaboração no desenvolvimento de sistemas colaborativos visando à elaboração de uma "Engenharia de Groupware" [9]. Dentre as ações realizadas para alcançar este objetivo, foi elaborado um processo específico para o desenvolvimento de sistemas colaborativos denominado RUP-3C-Groupware [19]. Este processo consiste numa extensão do Rational Unified Process [14] onde foram incorporadas as boas práticas aprendidas ao longo dos nove anos de experiência com a pesquisa e o desenvolvimento do AulaNet, que é um sistema colaborativo para o ensino-aprendizagem pela web [10]. O RUP-3C-Groupware documenta como o Modelo 3C de Colaboração é sistematicamente usado nas diferentes etapas do processo de desenvolvimento de um groupware: na análise de domínio para a classificação das aplicações groupware e seus elementos (seção 2); no foco a ser dado para o desenvolvimento de uma versão do groupware (seção 3), e na construção de componentes para a montagem do groupware (seção 4). Em relação ao desenvolvimento baseado em componentes [11], foram definidos kits de componentes 3C que o desenvolvedor usa para montar uma ferramenta de colaboração. Estas ferramentas, também classificadas em função do Modelo 3C, são usadas para compor sistemas colaborativos. Alguns resultados sobre as propostas apresentadas neste artigo são discutidos na seção 5 e a conclusão é apresentada na seção 6.

2. ANÁLISE DE GROUPWARE

Para analisar as aplicações colaborativas, um dos primeiros sistemas de classificação a ficar amplamente conhecido foi a taxonomia espaço-tempo [3]. Outra proposta foi a classificação dos sistemas colaborativos ao nível da aplicação, sendo enunciadas 7 classes de aplicações colaborativas [5]. Estas classes foram analisadas em função do grau de suporte à comunicação, coordenação e cooperação, sendo posicionadas no espaço triangular apresentado na Figura 1 [1, 22].

SBSC Simpósio Brasileiro de Sistemas Colaborativos

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 58-67.

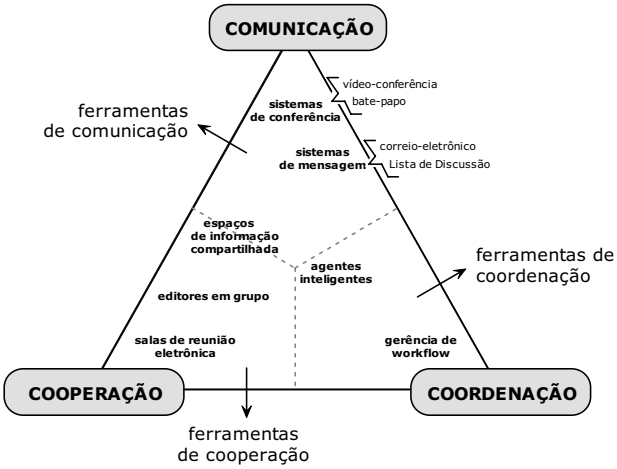


Figura 1. Classificação 3C dos sistemas colaborativos

Dividindo o espaço triangular em três seções, como apresentado na Figura 1, obtêm-se a classificação dos sistemas colaborativos em função do Modelo 3C. Este é o sistema de classificação adotado para a organização dos serviços de colaboração do ambiente AulaNet, apresentados na Figura 2.

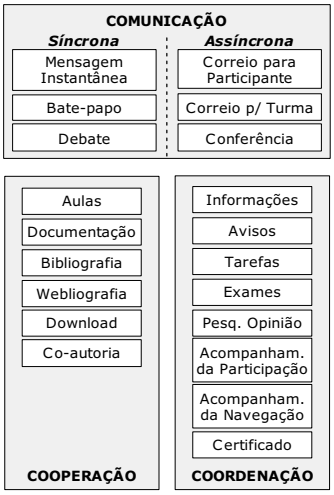


Figura 2. Serviços 3C do AulaNet 2.1

O AulaNet disponibiliza serviços que podem ser usados num curso. Os serviços são colocados à disposição do docente durante a criação e atualização de um curso, possibilitando-o selecionar quais vão se tornar serviços disponíveis aos aprendizes, configurando a área de trabalho do curso. Dessa maneira, o coordenador é levado a observar o grau de suporte que está sendo oferecido para a comunicação, coordenação e cooperação entre os aprendizes e mediadores do curso. Como exemplo, no curso TIAE [16], que objetiva promover aprendizagem colaborativa, são selecionados serviços relacionados a todos os C's, tais como: Debate e Conferência para promover discussão entre os aprendizes, Acompanhamento da Participação para coordenar a participação da turma; Tarefas para possibilitar aprendizagem baseada em projetos, e alguns serviços de cooperação para organizar os conteúdos trabalhados na turma. Por outro lado, quando o objetivo é elaborar um curso baseado na transmissão de informação (em vez aprendizagem colaborativa), então são selecionados serviços como Aulas para disponibilizar conteúdos (cooperação), Avisos e Exames (coordenação) para coordenar as atividades da turma e realizar a avaliação dos conteúdos assimilados por cada aluno – nenhum serviço de comunicação precisaria ser selecionado uma vez que não se objetiva estabelecer o diálogo entre os aprendizes (este é o método usado no ensino tradicional e que não promove a colaboração).

Apesar da separação com o propósito de análise, as dimensões comunicação, coordenação e cooperação não devem ser vistas de maneira isolada, pois são interdependentes. Um ambiente colaborativo, como exemplificado pelo AulaNet, deve dar suporte aos três C's. O posicionamento dos sistemas colaborativos no espaço triangular enfatiza a contigüidade entre os 3C's. Ainda que o objetivo de uma ferramenta possa voltar-se para o suporte específico de um dos C's, ainda assim contemplará aspectos dos demais C's. Por exemplo, um sistema de mensagem como o correio eletrônico, embora projetado para estabelecer a comunicação, também pode ser usado para distribuir ordens aos funcionários ou coordenar o trabalho de uma equipe, servindo assim para também dar suporte à coordenação. É por isso que, no espaço triangular, a classe Sistemas de Mensagem encontra-se perto do vértice Comunicação (principal objetivo), mas está deslocada na direção do vértice Coordenação (possibilita algum suporte, embora não seja este o seu principal objetivo).

Identifica-se que cada ferramenta contém mecanismos para dar suporte aos 3C's. Por exemplo, analisando uma ferramenta típica de bate-papo, Figura 3, são identificados três principais mecanismos: uma área para digitar a mensagem que possibilita o usuário se comunicar com os demais participantes, constituindo-se num suporte à Comunicação; uma lista de participantes indicando os que estão conectados e disponíveis para a conversação, constituindo-se num suporte à Coordenação; e uma área apresentando o registro das mensagens enviadas, constituindo-se num suporte à Cooperação. A ferramenta de bate-papo é um bom exemplo para evidenciar que as ferramentas de colaboração contêm elementos relacionados aos 3 C's. Mesmo sendo uma ferramenta de comunicação, pois seu objetivo é possibilitar a troca de mensagens entre os membros de um grupo, uma ferramenta de bate-papo também contém elementos de coordenação e cooperação que são necessários para organizar e documentar a comunicação.

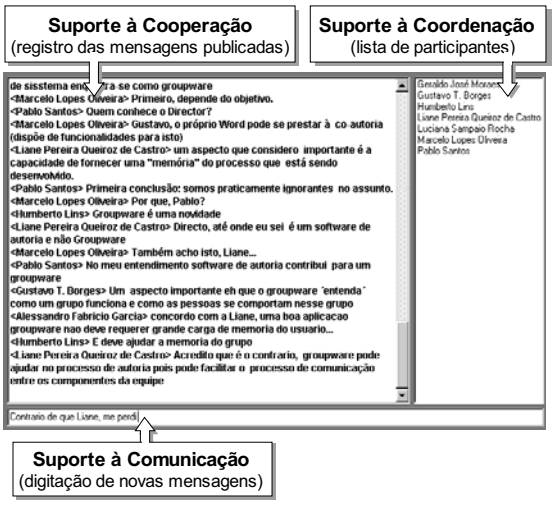


Figura 3. Elementos 3C de um sistema de bate-papo

Seguindo este tipo de análise, é possível identificar os elementos constituintes de uma família de aplicação classificando-os em função do Modelo 3C, tal como apresentado no Quadro 1 para o domínio das ferramentas de comunicação síncrona.

Quadro 1. Classificação 3C dos elementos das ferramentas de comunicação síncrona

Comunicação	Linguagem	Principais linguagens da comunicação humana: escrita (texto), falada (áudio), pictórica (imagens e animações) e gestual (vídeo e avatar).
	Transmissão	Pontual (após o emissor formular toda a mensagem) ou Contínua (transmissão contínua de vídeo e áudio, ou caracter-a-caracter enquanto a mensagem está sendo formulada)
	Tamanho e Qualidade	Tamanho: quantidade de caracteres (texto) ou a duração em segundos (vídeo e áudio). Qualidade do vídeo e do áudio é geralmente reduzida para a transmissão pela Internet
	Estruturação do discurso	Estrutura linear (uma mensagem apresentada após a outra), hierárquica (árvore, *threads*) ou em rede (grafo, mapas)
	Categorização	Rótulos para caracterizar as mensagens, tais como: tipo de fala (sussurra, pergunta, concorda, etc.); tipo de discurso (direto ou indireto), tipo de emoção (alegre, zangado) etc.
Coordenação	Tópico	Assunto a ser discutido
	Sessão	Espaço de tempo para a duração da conversação
	Acesso	Quem ou quantos podem participar da conversação
	Presença	Quem está participando da conversação
	Disponibilidade	Status do participante: presente, ausente, ocupado, etc.
	Papéis	Atribuição de papéis: Operador, Mediador, Moderador etc.
	Posse da palavra	Quem pode falar num dado momento
	Freqüência	Limite da quantidade de mensagens que podem ser enviadas num intervalo de tempo
	Visibilidade	Pública (visível para todos os participantes) ou particular (restrita a dois participantes)
	Endereçamento	Indicação do destinatário da mensagem
	Indicação do turno-em-desenvolvimento	Informação de que o participante está formulando a mensagem (antes de sua transmissão pontual)
	Avaliação	Qualificação das mensagens, dos participantes ou da sessão
Cooperação	Registro	Armazenamento das mensagens publicadas
	Configuração do espaço	Visualização e Recuperação das mensagens publicadas
	Mensagens preconcebidas	Mensagens pré-elaboradas disponíveis para os participantes trocarem durante a conversação

O Modelo 3C de Colaboração é usado por nosso grupo de pesquisa como um guia para analisar um domínio de aplicação groupware. Os procedimentos para realizar esta análise foram documentados no detalhe de fluxo "Analisar Domínio" do RUP-3C-Groupware, conforme esquematizado na Figura 4 (em conformidade com a notação do RUP padrão).

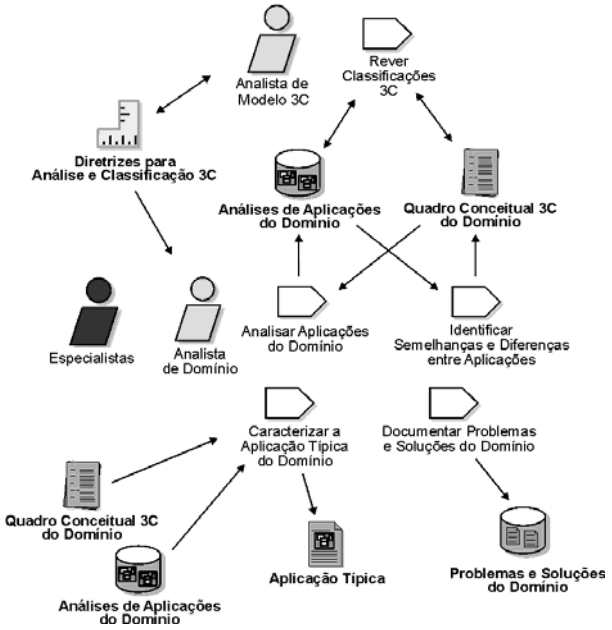


Figura 4. Detalhe de fluxo "Analisar Domínio" proposto na disciplina Modelagem de Negócio do RUP-3C-Groupware

De acordo com este detalhe de fluxo, cabe ao Analista de Domínio analisar diferentes aplicações do domínio para o qual o novo groupware está sendo desenvolvido. Este analista deve consultar diversas fontes de informação tais como especialistas do domínio, aplicações existentes, publicações e outros documentos [23]. O analista estabelece comparações entre as aplicações buscando abstrair os elementos de comunicação, coordenação e cooperação do domínio. Como resultado desta atividade, objetiva-se construir um Quadro Conceitual 3C do domínio, ou aperfeiçoar algum que já esteja em uso no projeto. Ao analisar as aplicações do domínio, deve-se documentar as principais funcionalidades classificando-as de acordo com o Quadro Conceitual 3C. O analista também deve caracterizar o que é uma aplicação típica daquele domínio, identificando seu conjunto mínimo e relevante de elementos, o que servirá como base para o desenvolvimento de sucessivas versões do groupware em busca da solução de problemas. Alguns problemas e suas soluções naquele domínio já podem ser conhecidos e devem estar documentados num repositório, tornando-se útil para auxiliar o analista na seleção ou especificação de uma variação de solução que já se saiba ser adequada ao menos em outras aplicações. Deve-se, ainda, contar com um Analista de Modelo 3C que será responsável pelo uso consistente deste modelo ao longo do processo de desenvolvimento do groupware.

3. DESENVOLVIMENTO ITERATIVO E INVESTIGATIVO FOCANDO UM PROBLEMA E UM "C" POR VERSÃO

Desenvolver software, especialmente groupware, é resolver problemas. Geralmente, um projeto de groupware inicia porque as aplicações existentes não satisfazem as necessidades de um grupo, sendo identificado um conjunto de problemas que se deseja resolver. Uma boa prática é tentar resolver um problema por vez. A cada versão, seleciona-se um problema específico para o qual se projeta uma solução a partir da qual são derivados os requisitos da versão. Quando a versão do groupware tiver sido construída, desenvolve-se um estudo de caso para avaliar em que medida a implementação da solução mostra-se adequada na resolução do problema. A partir da análise de dados coletados do estudo de caso, pode-se concluir se a versão está suficientemente adequada para ser liberada para o uso, ou então, identificar modificações que precisam ser feitas ou novos problemas que ainda precisam ser resolvidos, dando início a um novo ciclo de desenvolvimento. Esta é a prática aprendida por nosso grupo de pesquisa para o desenvolvimento de groupware: desenvolvimento interativo focando a resolução de um problema por versão.

O Modelo 3C de Colaboração tem se mostrado útil para guiar o estabelecimento do foco a ser dado no desenvolvimento de cada versão. Cada versão da aplicação groupware é desenvolvida para resolver ora um problema de comunicação, ora de coordenação, ora de cooperação, conforme esquematizado na Figura 5. O desenvolvimento em função do Modelo 3C de Colaboração ajuda a prever que dimensão da colaboração deve ser observada em função da modificação de um determinado elemento, auxilia o projeto da aplicação e a análise dos resultados obtidos de estudos de caso.

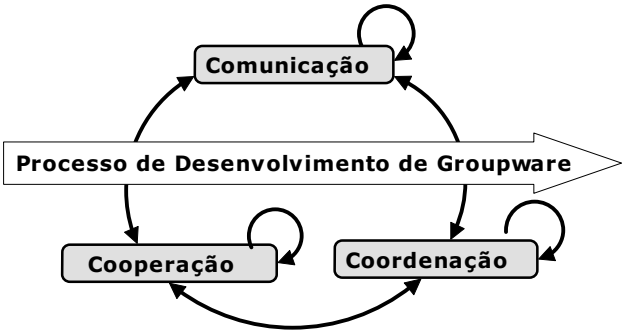


Figura 5. Processo de Desenvolvimento baseado no Modelo 3C de Colaboração

Esta prática é exemplificada com o desenvolvimento das versões do Mediated Chat [7], Figura 6. Estas versões têm sido desenvolvidas para adequar a ferramenta de bate-papo ao uso educacional. A Confusão do Bate-papo foi a principal limitação identificada sobre o uso do bate-papo na educação. A Confusão do Bate-papo é potencializada por um conjunto de problemas sobrepostos. Procurou-se focar na busca pela solução de um único problema específico no desenvolvimento de cada versão do Mediated Chat, o que possibilitou compreender melhor o problema e a solução implementada, e possibilitou identificar novos problemas que ainda precisavam ser resolvidos.

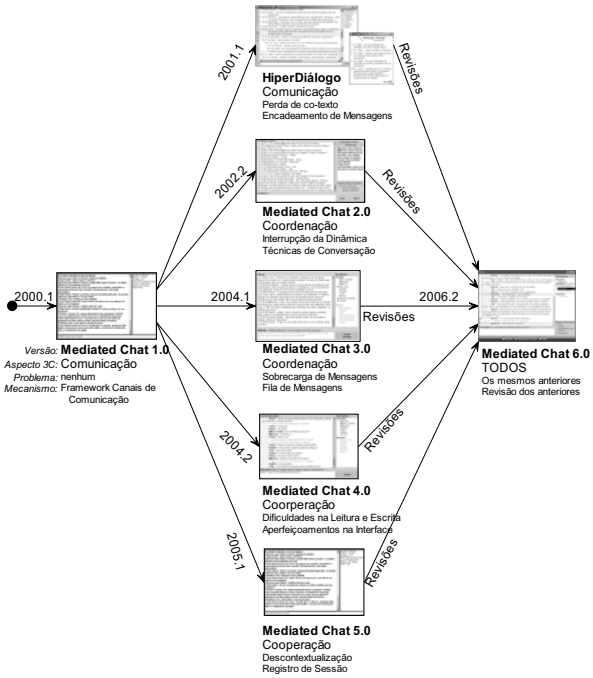


Figura 6. Desenvolvimento das versões Mediated Chat

O desenvolvimento em sucessivas versões é especialmente útil para os sistemas colaborativos uma vez que mudanças na ferramenta geram modificações por vezes imprevistas e indesejáveis na maneira do grupo trabalhar, sendo adequado rever a solução implementada numa versão seguinte. O desenvolvimento da versão Mediated Chat 2.0 exemplifica este desencadeamento de modificações e revisões. Para diminuir o problema da Confusão do Bate-papo, foi elaborada uma dinâmica mais estruturada de conversação. Ao aplicar a dinâmica, foi observada a ocorrência de um novo problema: os participantes frequentemente enviavam mensagens inadequadas à etapa de conversação em vigor, causando a Interrupção da Dinâmica. A ferramenta Mediated Chat 2.0 foi então desenvolvida com um conjunto de técnicas de conversação para definir quem pode falar a cada instante (posse da palavra). Ao realizar um novo estudo de caso, constatou-se que as técnicas implementadas não possibilitavam contornar situações excepcionais, gerando novos problemas. A implementação foi então revisada e integrada na versão Mediated Chat 6.0.

Para o desenvolvimento de groupware, deve-se focar num dos C's por versão. Esta estratégia tem guiado o desenvolvimento das versões Mediated Chat. Por exemplo, na versão Mediated Chat 1.0, o foco foi no estabelecimento da Comunicação; já na versão 3.0, o foco relaciona-se com a Coordenação. Na versão 3.0, procurou-se resolver o problema denominado "Sobrecarga de Mensagem", que ocorre quando várias mensagens são enviadas num curto intervalo de tempo, o que inviabiliza a leitura de todas as mensagens potencializando a confusão da conversação. A solução implementada nesta versão foi a "Fila de Publicação", onde o servidor aguarda um intervalo de tempo após publicar uma mensagem. O intervalo de tempo é proporcional à quantidade de caracteres, tendo sido empiricamente estimado como suficiente para a leitura da mensagem recém publicada. Durante este intervalo de tempo, as novas mensagens enviadas pelos participantes são enfileiradas no servidor para a posterior

publicação. Este mecanismo distribui a publicação das mensagens ao longo do tempo de tal forma que os participantes consigam ler todas as mensagens sem serem surpreendidos por rajadas de mensagens.

Nem sempre é possível alterar apenas um dos C's por versão. Ao alterar um dos aspectos da colaboração, pode ser necessário alterar também um outro aspecto. Por exemplo, ainda sobre a versão Mediated Chat 3.0, com o uso da "Fila de Mensagem", uma mensagem com um texto muito longo faria o servidor ficar suspenso por muito tempo tornando o bate-papo moroso. Para evitar este novo problema, foi estabelecida uma quantidade máxima de caracteres por mensagem. Essa limitação relaciona-se com a Comunicação (ver Quadro 1, elemento "tamanho"). Portanto, apesar desta versão focar na resolução de um problema relacionado à Coordenação, foi necessário modificar também um elemento relacionado à Comunicação. É preciso estar ciente que os C's são interdependentes, e alterar um pode requerer a alteração de outro.

Também não é necessário resolver um único problema por versão. Ainda tendo como exemplo a versão Mediated Chat 3.0, ao projetar funcionalidades para indicar quem está na fila e quem está falando, também se projetou a indicação de quem está digitando, pois são os participantes que possivelmente entrarão na fila de publicação. Este mecanismo acabou resolvendo outro problema: Falta-de-visibilidade-do-turno-em-desenvolvimento. Nesta versão, foi possível lidar com a solução destes dois problemas, embora tenha ficado mais difícil inferir as influências dos mecanismos implementados.

A boa prática "focar um único problema e um único 'C' por versão" é uma diretriz para nortear o desenvolvimento, mas não deve ser seguida invariavelmente. Pode ser que a solução introduza novos problemas sendo preciso modificar outros elementos, inclusive de natureza diferente do C originalmente focado. Pode ser que numa implementação coerente sejam resolvidos dois problemas ao mesmo tempo. Idealmente, para se realizar uma pesquisa rigorosa, em cada versão deve-se projetar uma solução que modifique apenas um elemento do domínio para facilitar a análise de suas influências. Na prática, este diretriz deve ser aplicada com bom senso.

O objetivo de se focar num único C é induzir o projetista a observar os aspectos da colaboração questionando-se de que maneira o novo mecanismo implementado estará influenciando a comunicação, a coordenação e a cooperação. A organização dos elementos do domínio em função do Modelo 3C de Colaboração ajuda a prever mudanças e auxilia na análise dos resultados obtidos de estudos de caso.

Num processo de desenvolvimento de groupware, a prática "desenvolvimento iterativo e investigativo focando um problema e um C por versão" se realiza nas etapas Requisitos e Testes.

Figura 7. Principais atividades especificadas no detalhe de fluxo "Analisar Problema" da disciplina Requisitos do RUP-3C-Groupware

No RUP padrão, os requisitos do sistema são documentados no artefato Visão durante a atividade Desenvolver Visão que faz uso das Regras de Negócio e das Solicitações do Interessado. No RUP-3C-Groupware, como ilustrado na Figura 7, primeiro deve-se executar a atividade Isolar Problema para documentar os problemas relatados pelos interessados. Deve-se consultar o repositório de Problemas e Soluções do Domínio para comparar os problemas relatados com outros já conhecidos. Se for identificado um novo problema, deve-se produzir o artefato Documentação de Problema e cadastrá-lo no repositório. Formalizados os problemas relatados pelo interessado, deve-se selecionar um único problema (o de mais alta prioridade) para desenvolver uma nova versão do groupware. Na atividade Desenvolver Visão, o artefato visão a ser produzido deve ser derivado do problema selecionado. Para especificar as características da versão a ser desenvolvida, que serão usadas para definir os Requisitos, deve-se partir de uma solução já conhecida, se existir, ou propor uma variação de alguma solução relacionada. Ao estabelecer as características da versão a ser desenvolvida a partir da solução proposta, deve-se considerar as outras Aplicações do Domínio, o Quadro Conceitual 3C, e os elementos já existentes na Aplicação Típica (caso seja a primeira versão a ser desenvolvida) ou numa versão desenvolvida anteriormente.

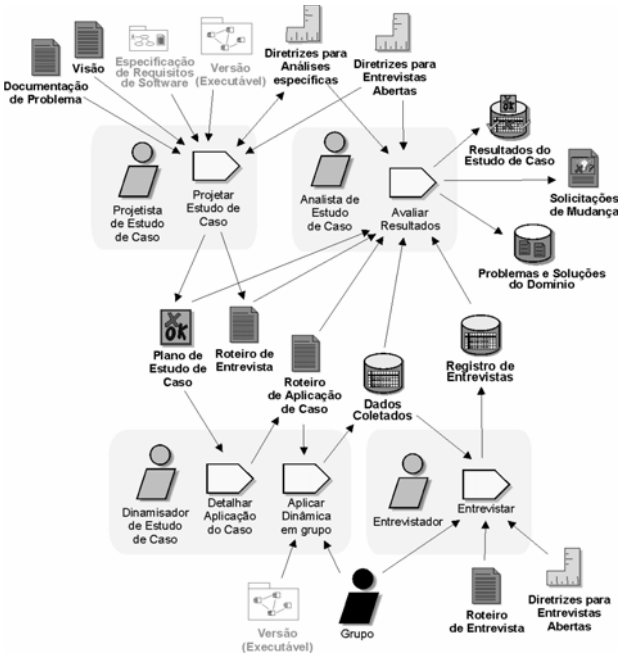


Figura 8. Detalhes do fluxo "Realizar Estudo de Caso" do RUP-3C-Groupware

Na disciplina Teste, procura-se por erros de implementação e inferir a conformidade com os requisitos. No RUP-3C-Groupware, foi especificado um teste de validação para investigar a adequação da solução proposta para o problema que originou o desenvolvimento da versão, o que é realizado através do fluxo Estudo de Caso apresentado na Figura 8. Embora a realização de estudo de caso seja uma prática mais específica para o desenvolvimento de groupware, esta etapa não é influenciada diretamente pelo Modelo 3C de Colaboração, mas sim por princípios da pesquisa etnográfica. Este é o detalhe de fluxo que possibilita o desenvolvimento investigativo de groupware.

4. COMPONENTES 3C

Um sistema colaborativo geralmente integra um conjunto de ferramentas para colaboração. O Quadro 2 lista diferentes ferramentas disponíveis em diferentes sistemas colaborativos. Alguns tipos de ferramenta são usados em diferentes sistemas colaborativos – por exemplo, a maioria dos sistemas analisados oferece fórum, bate-papo, agenda, relatórios de atividades, questionários, gerenciamento de tarefas, votação, repositório e links. Cada ferramenta pode ser vista de forma relativamente autônoma dentro do sistema colaborativo. Estas características são propícias à aplicação de técnicas de desenvolvimento baseado em componentes, onde as ferramentas são componentes do sistema colaborativo a serem instanciados e configurados.

Quadro 2. Ferramentas 3C dos ambientes Colaborativos

		Ambientes Colaborativos								
		AulaNet	TelEduc	AVA	WebCT	Moodle	GroupSystems	YahooGroups	OpenGroupwar	BSCW
C	Correio	x	x	x	x				x	
	Lista de Discussão	x		x				x		
	Fórum	x	x	x	x	x	x			x
	Mural		x	x					x	
	Brainstorming						x			
	Bate-papo (chat)	x	x	x	x	x		x		
	Mensageiro	x				x				
Coordenação	Agenda		x	x	x	x	x	x	x	x
	Relat de Atividades	x	x	x	x	x		x		x
	Acomp. da Particip.	x	x	x		x	x			
	Questionário	x	x	x	x	x	x	x		
	Tarefas	x	x	x	x	x	x		x	x
	SubGrupos	x	x			x	x			x
	Gerenc. de recursos								x	
	Orientação			x						
	Votação	x			x	x	x	x		x
Coordenação	Conteúdos	x	x	x	x	x		x		x
	Quadro Branco				x					
	Busca				x	x		x		x
	Glossário			x	x	x		x		
	Links	x	x	x	x	x		x		x
	Jornal Cooperativo					x				
	Classificador						x			
	Wiki					x				
	Gerenc. de contatos								x	x
	Revisão em pares					x				
	FAQ		x	x		x				
	Anotações	x	x	x	x		x			
	RSS					x		x		

Além dos sistemas colaborativos possuírem ferramentas similares, tais ferramentas também possuem funcionalidades similares. Por exemplo, os serviços Conferências e Correio para Turma do ambiente AulaNet compartilham o suporte ao envio, ao recebimento e à exibição de mensagens, à categorização, à avaliação da participação e ao bloqueio do canal de comunicação, entre outras funcionalidades. Encapsular as funcionalidades recorrentes em componentes propicia também o reuso do suporte computacional à colaboração, aumentando o reuso de código. Passa também a ser possível evoluir, ajustar e construir serviços variando e reconfigurando os componentes de colaboração.

Estes cenários indicam a utilidade de se adotar o desenvolvimento de groupware baseado em dois níveis de componentes. O primeiro nível contempla os componentes que provêem os serviços colaborativos, usados para oferecer suporte computacional à dinâmica da colaboração como um todo. O segundo nível contempla os componentes usados para montar ferramentas de colaboração, oferecendo suporte a determinados aspectos da colaboração dentro de uma ferramenta em particular. Nesta

abordagem proposta, como ilustrado na Figura 9, os componentes que implementam as ferramentas colaborativas são chamados de *serviços* e os componentes usados para implementar o suporte computacional à colaboração dos serviços são chamados de *componentes de colaboração*. Mesmo um serviço de comunicação, como uma ferramenta de bate-papo, além dos componentes de comunicação, também usa componentes de coordenação e de cooperação. Os componentes de colaboração de um C são reusados nos serviços dos demais C's.

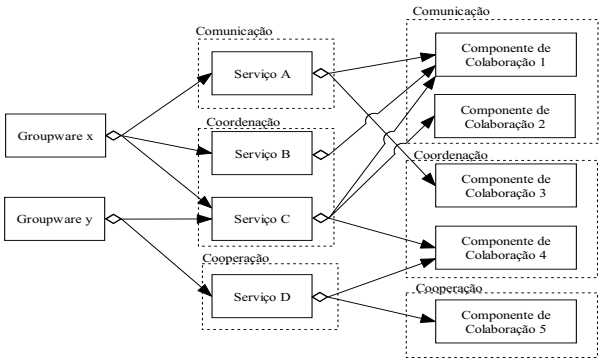


Figura 9. Sistemas colaborativos montados a partir de serviços, e serviços montados a partir de componentes de colaboração

O Modelo 3C de Colaboração é útil nesta abordagem para definir uma sistemática de classificação para os componentes. A partir de kits de componentes de colaboração, organizados em função do Modelo 3C, o desenvolvedor monta um serviço. Cada serviço, por sua vez também classificado em função do Modelo 3C, pode ser usado para montar um sistema colaborativo. Além do reuso, esta abordagem favorece também a capacidade de extensão da solução ao possibilitar a inclusão de novos componentes. As subseções a seguir detalham esta abordagem para a montagem de groupware a partir de componentes 3C.

4.1 Collaboration Component Kit

Um *component kit* é um conjunto de componentes interoperáveis aderentes a uma padronização. De um *component kit* gera-se uma família de aplicações, fazendo diferentes arranjos e eventualmente desenvolvendo alguns sob medida [24]. Para desenvolver um *component kit*, são analisadas aplicações similares e são identificados e generalizados componentes comuns [4]. Os *component kits* são extensíveis para acomodar novos componentes quando necessários. Componentes de software que sejam realmente reusáveis são refinados iterativamente até atingir a maturidade, a confiabilidade e a adaptabilidade desejadas [13].

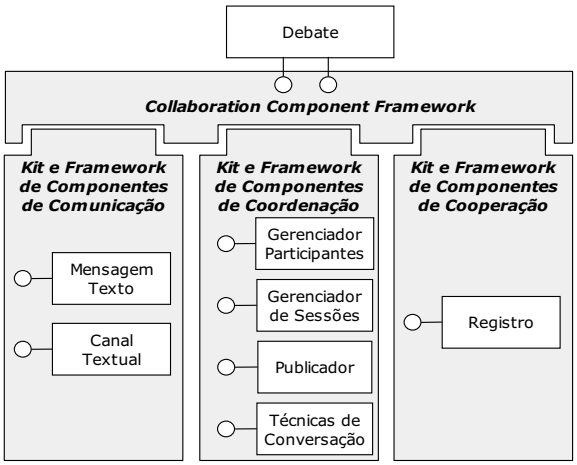


Figura 10. Montagem do serviço Debate a partir dos kits de componentes baseados no Modelo 3C de Colaboração

Os componentes do Collaboration Component Kit são usados pelo desenvolvedor de groupware na composição dos serviços colaborativos. A Figura 10 ilustra a construção do serviço Debate no AulaNet 3.0. Para a montagem deste serviço foram selecionados alguns dos componentes 3C do *Collaboration Component Kit*.

4.2 Arquitetura Componentizada

Para oferecer suporte ao gerenciamento e à execução dos componentes, são usados *component frameworks* [21]. Um *component framework* é um conjunto de interfaces e regras de interação que possibilitam a implantação de componentes aderentes a um padrão. Na arquitetura proposta, é feito uso de um *component framework* para cada tipo de componente proposto. No Service Component Framework são acoplados os serviços, oferecendo suporte à montagem do sistema colaborativo, e no Collaboration Component Framework são acoplados os componentes de colaboração, usados na montagem do serviço. A Figura 11 ilustra a implantação dos componentes nos *component frameworks* correspondentes, que estabelecem as condições ambientais para as instâncias dos componentes e oferecem serviços relativos ao ciclo de vida.

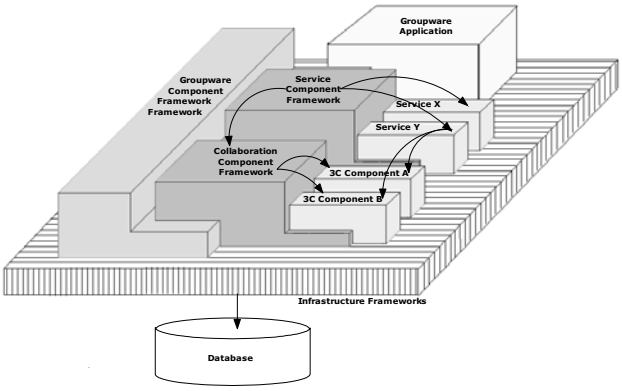


Figura 11. A arquitetura de aplicação proposta

Os *component frameworks* são responsáveis por tratar a instalação, remoção, atualização, ativação, desativação,

localização, configuração, monitoramento, importação e exportação de componentes. O Service Component Framework gerencia as instâncias dos serviços e a ligação com os componentes de colaboração correspondentes. O Collaboration Component Framework gerencia as instâncias dos componentes de colaboração, que são provenientes do Collaboration Component Kit.

Grande parte das funcionalidades dos *component frameworks* é recorrente e reusável. Um framework pode ser usado para a instanciação de uma família de sistemas. Na arquitetura proposta, é utilizado um framework para instanciar os *component frameworks*. Este tipo de framework é chamado de *component framework framework* (CFF) [21]. Um *component framework framework* é visto como um *component framework* de segunda ordem, onde seus componentes são *component frameworks*. Da mesma forma que um componente interage com outros diretamente ou mediado pelo *component framework*, o mesmo pode ser dito dos *component frameworks*, cujo suporte de mais alto nível é o *component framework framework*. A Figura 11, estendendo a notação utilizada por Szyperski [21], ilustra a arquitetura de aplicação, incluindo o Groupware Component Framework Framework, como o *component framework* de segunda ordem.

A arquitetura elaborada segue uma divisão em camadas, importante para tratar a complexidade de sistemas componentizados. A camada de apresentação (não representada na Figura 11) é responsável pela captura e apresentação de dados e pela interação com o usuário; a camada de negócio captura o modelo da lógica de negócio do domínio da aplicação; e a camada de infra-estrutura implementa os serviços técnicos de baixo nível. A mesma infra-estrutura desenvolvida para a camada de negócio pode ser usada para mais de uma apresentação, como por exemplo, PDA, desktop implementado em HTML e desktop implementado em Flash (*Rich Internet Application*). Quando os serviços da camada de negócio necessitam de acesso remoto, como por exemplo um cliente PDA, são disponibilizados web services que encapsulam a fachada da camada de negócio. Nos demais casos, a apresentação acessa diretamente a fachada do negócio.

O ferramental desenvolvido com esta pesquisa instrumenta o desenvolvimento da camada de negócio, implementando os conceitos do Modelo 3C de Colaboração. A arquitetura de aplicação expressa a estrutura dos componentes do domínio, representando um projeto lógico de alto nível independente da tecnologia de suporte [4]. Instrumentado pelo Modelo 3C de Colaboração, o desenvolvedor modela a aplicação e seus requisitos, e seleciona os serviços e seus componentes de modo a oferecer suporte às necessidades de colaboração. O desenvolvedor seleciona os componentes desejados a partir dos *component kits*, implantando-os nos *component frameworks* correspondentes.

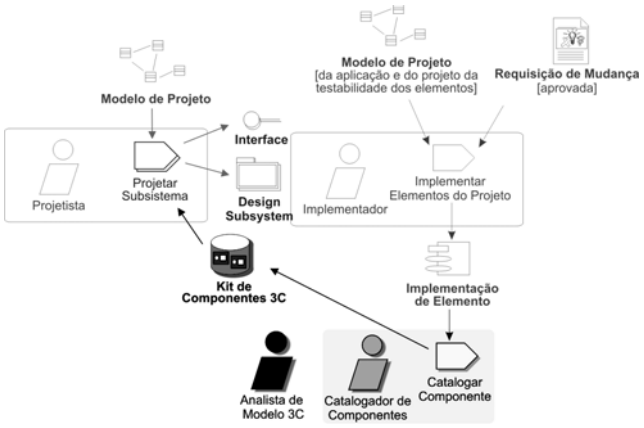


Figura 12. Uso do Kit de Componentes 3C no RUP-3C-Groupware

Num processo sistemático de desenvolvimento, a abordagem baseada em componentes influencia as disciplinas de Projeto e Implementação. Como ilustrado na Figura 12, no RUP-3C-Groupware, o projetista deve levar em consideração os componentes de colaboração já existentes ao Projetar Subsistema. Quando necessário, novos componentes deverão ser projetados, implementados e catalogados no repositório Kit de Componentes 3C.

5. RESULTADOS OBTIDOS

Para obter indícios se engenheiros de software (além dos que constituem a equipe de desenvolvimento do Projeto AulaNet e do grupo de pesquisa Groupware@LES) conseguiriam usar o Modelo 3C Colaboração no desenvolvimento de groupware, foram realizados Estudos de Caso com os alunos de graduação (2 alunos) e pós-graduação (3 de mestrado e 2 de doutorado) da disciplina Engenharia de Groupware do Departamento de Informática da PUC-Rio durante o segundo semestre de 2005. Dentre os trabalhos integrantes das atividades desta disciplina, individualmente, os alunos tiveram que selecionar um sistema colaborativo e analisar suas funcionalidades, classificando-as em comunicação, coordenação ou cooperação. Numa segunda etapa do trabalho, tiveram que efetuar um estudo comparativo de aplicações colaborativas similares à escolhida, propondo aprimoramentos através de incorporação de serviços e funcionalidades que estendam e complementem os três C's analisados do sistema. Numa terceira parte do trabalho, cada aluno apresentou uma arquitetura e um protótipo da extensão do sistema usando a infra-estrutura e os componentes 3C apresentados neste artigo. Por fim, os alunos executaram algumas das atividades-chave e produziram alguns artefatos-chave do processo RUP-3C-Groupware.

Os resultados dos estudos de caso foram avaliados por observação direta dos mediadores do curso e pela aplicação de questionários individuais. Dada a inexperiência da turma em lidar com a tecnologia de componentes e com o modelo de colaboração, os 76% de acerto foi considerado satisfatório. Os alunos receberam e responderam um questionário sobre o ferramental disponibilizado. A maior parte dos alunos avaliou o grau de dificuldade da utilização do modelo 3C para a análise da aplicação escolhida como regular, em uma escala de muito difícil a muito fácil. O entendimento do modelo 3C teve a mesma avaliação. Estes

resultados foram considerados satisfatórios, visto que os alunos tiveram o primeiro contato com o modelo 3C e com a abordagem durante o curso e não são especialistas em groupware. Com relação à abrangência do modelo 3C na modelagem do sistema, 5 alunos avaliaram como suficiente e 2 como regular, indicando que a maior parte das funcionalidades identificadas foram classificadas. Com relação à utilização dos componentes 3Cs, 5 alunos identificaram a solução como complexa e 2 como normal, em uma escala de muito simples a muito complexa. Este resultado também foi considerado satisfatório, dado que, além de não serem especialistas em groupware, os alunos não são especialistas em componentes de software. Apesar de a maioria ter classificado a solução como complexa, todos avaliaram a utilização de componentes 3C na composição de groupware como bom ou muito bom. Com relação ao encapsulamento das complexidades de baixo nível, 3 alunos avaliaram como neutro, 2 como bom e 2 como muito bom. Por fim, ao avaliar o suporte computacional a groupware utilizando componentes 3C, 2 alunos avaliaram como neutro e 5 como bom.

Em relação ao uso do processo RUP-3C-Groupware, ao final do trabalho os alunos deveriam preencher um Questionário avaliando as atividades e artefatos experimentados, e cada aluno foi entrevistado durante 15 minutos de acordo com o método de entrevista de perguntas abertas [18]. Os participantes julgaram todos os artefatos como sendo muito relevantes para o processo de desenvolvimento, e que são de entendimento e execução com grau de dificuldade médio para fácil. Além do julgamento subjetivo dos participantes, foi analisada a qualidade dos artefatos produzidos, e constatou-se que os alunos conseguiram produzir adequadamente a maioria dos artefatos-chave específicos do processo elaborado. Estes resultados indicam a repetitividade do processo: Engenheiros de Software conseguem seguir o RUP-3C-Groupware.

6. CONCLUSÃO

Este artigo mostrou o uso do Modelo 3C de Colaboração em diferentes etapas do desenvolvimento de groupware: modelagem de Negócio, Requisitos, Análise, Projeto e Implementação. Nosso grupo de pesquisa tem mostrado a utilidade deste modelo no processo de desenvolvimento de groupware [19, 8], no desenvolvimento de ferramentas de colaboração [7, 12], no desenvolvimento da versão AulaNet 3.0 (Figura 13.a) [20] e na versão AulaNetM para equipamentos móveis (Figura 13.b) [6].

a) AulaNet 3.0 para Desktop

b) AulaNetM para PDA e Celular

Figura 13. Novas versões do AulaNet

O processo RUP-3C-Groupware encontra-se em sua versão inicial e, a médio prazo, novas pesquisas devem ser realizadas buscando a melhoria contínua do processo elaborado. Mesmo em sua versão inicial, este processo já é útil para auxiliar o desenvolvimento de groupware, fornecendo diretrizes para: usar o Modelo 3C de Colaboração na análise e desenvolvimento do groupware; desenvolver versões do groupware focando um problema por versão, num processo evolucionário e investigativo que inclui a realização de estudos de caso; e desenvolver groupware fazendo uso da abordagem baseada em componentes e orientada ao reuso.

7. AGRADECIMENTOS

O Projeto AulaNet é parcialmente financiado pela Fundação Padre Leonel Franca e pelo Ministério da Ciência e Tecnologia através do projeto Sistemas Multi-Agentes para a Engenharia de Software (ESSMA) bolsa nº 552068/2002-0. Também é financiado pelas bolsas individuais do CNPq: Carlos Lucena nº 30091/2003-6, Hugo Fuks nº 301917/2005-1, Marco Aurélio Gerosa nº 383719/2006-2 e Mariano Pimentel nº 383718/2006-6. Denise Filippo recebe bolsa CCPG/VRAc da PUC-Rio.

8. REFERÊNCIAS

[1] Borghoff, U.M. & Schlichter, J.H. Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, USA, 2000.

[2] Calvary, G., Coutaz, J. & Nigay, L. From Single-User Architectural Design to PAC*: a Generic Software Architectural Model for CSCW. Conference on Human Factors in Computing Systems (CHI'97), 1997. pp 242-249.

[3] DeSanctis, G., Gallupe, B.. A foundation for the study of group decision support systems. Management Science, v. 33, n. 5. 1987. p. 589-609.

[4] D'Souza, D.F., Wills, A.C. Objects, Components and Frameworks with UML: The Catalysis Approach. Addison Wesley, 1998.

[5] Ellis, C.A., Gibbs, S.J. & Rein, G.L. Groupware - Some Issues and Experiences. In: Communications of the ACM, v. 34, n. 1. 1991, p. 38-58.

[6] Filippo, D., Fuks, H., & Lucena, C.J.P. AulaNetM: Extensão do Serviço de Conferências do AulaNet destinada a usuários de PDAs. Anais do XVI Simpósio Brasileiro de Informática na Educação - SBIE 2005, Juiz de Fora, MG, 9 a 11 de novembro de 2005, pp. 623-633.

[7] Fuks, H., Pimentel, M., Lucena, C.J.P. R-U-Typing-2-Me? Evolving a chat tool to increase understanding in learning activities. International Journal of Computer-Supported Collaborative Learning, v. 1, n. 1, ISSN 1556-1607 (Paper) 1556-1615 (Online). Springer, 21 Março 2006. p 117-142. Também disponível online: http://dx.doi.org/10.1007/s11412-006-6845-3

[8] Fuks, H., Raposo, A., Gerosa, M.A. & Lucena, C.J.P. Applying the 3C Model to Groupware Development. International Journal of Cooperative Information Systems (IJCIS), v.14, n.2-3, World Scientific, Jun-Sep 2005. pp. 299-328.

[9] Fuks, H., Raposo, A. & Gerosa, M.A. Do Modelo de Colaboração 3C à Engenharia de Groupware. WEBMIDIA 2003 - Simpósio Brasileiro de Sistemas Multimídia e Web, Trilha especial de Trabalho Cooperativo Assistido por Computador, Novembro 2003, Salvador-BA, pp. 445-452.

[10] Fuks, H., Gerosa, M.A., Lucena, C.J.P. The Development and Application of Distance Learning on the Internet. Open Learning - The Journal of Open and Distance Learning, v. 17, n. 1, 2002. p. 23-38.

[11] Gerosa, M.A. Desenvolvimento de Groupware Componentizado com Base no Modelo 3C de Colaboração. Tese de Doutorado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), 16 de março de 2006.

[12] Gerosa, M.A., Pimentel, M., Filippo, D., Barreto, C.G., Raposo, A., Fuks, H. & Lucena, C.J.P. Componentes Baseados no Modelo 3C para o Desenvolvimento de Ferramentas Colaborativas. Anais do 5º Workshop de Desenvolvimento Baseado em Componentes - WDBC 2005, 7-9 de novembro de 2005, Juiz de Fora, MG, pp. 109-112.

[13] Gimenes, I.M.S., Huzita, E.H.M (orgs.) Desenvolvimento Baseado em Componentes: Conceitos e Técnicas. Rio de Janeiro: Ciência Moderna, 2005.

[14] Kruchten, P. Introdução ao RUP – Rational Unified Process. Rio de Janeiro: Ciência Moderna, 2003.

[15] Laurillau, Y. & Nigay, L. Clover architecture for groupware. Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 2002), 2002. pp. 236 – 245

[16] Lucena, C.J.P., Fuks, H. Tecnologias de Informação Aplicadas à Educação (TIAE): Manual do Aprendiz. Monografias em Ciência da Computação, no 07/02, Departamento de Informática, PUC-Rio, 2002. Também disponível online: http://groupware.les.inf.puc-rio.br/courses/Manual_Aprendiz_TIAE.pdf

[17] Neale, D.C., Carroll, J.M. & Rosson, M.B. Evaluating computer-supported cooperative work: models and frameworks, Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04), Chicago, Illinois, USA, November 06 - 10, ACM Press, New York, 2004. pp. 112-121.

[18] Nicolaci-da-Costa, A. M., Leitão, C. F. e Romão-Dias, D. Gerando conhecimento sobre homens, mulheres e crianças que usam computadores: algumas contribuições da psicologia clínica, IV Workshop sobre Fatores Humanos em Sistemas Computacionais, Florianópolis, 2001.

[19] Pimentel, M. RUP-3C-Groupware: um processo de desenvolvimento de groupware baseado no Modelo 3C de Colaboração. Tese de Doutorado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), 22 de março de 2006.

[20] Pimentel, M., Gerosa, M.A., Filippo, D., Barreto, C.G., Raposo, A., Fuks, H. & Lucena, C.J.P. AulaNet 3.0: desenvolvendo aplicações colaborativas baseadas em componentes 3C. WCSCW 2005 - Workshop Brasileiro de Tecnologias para Colaboração, 7 e 8 de Novembro 2005. Em Anais XVI Simpósio Brasileiro de Informática na Educação, v. 2, ISBN 85-88279-48-7. Juiz de Fora - MG: UFJF, 8 a 11 de Novembro 2005. p. 761-770.

[21] Szyperski, C. Component Software: Beyond Object-Oriented Programming, Addison-Wesley, 1997.

[22] Teufel, S., Sauter, C., Mühlherr, T., Bauknecht, K. Computerunterstützte Gruppenarbeit. Bonn: Addison-Wesley, 1995 apud Borghoff, U.M. and Schlichter, J.H., Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, USA, 2000.

[23] Werner, C.M.L., Braga, R.M.M. A Engenharia de Domínio e o Desenvolvimento Baseado em Componentes. Desenvolvimento Baseado em Componentes: Conceitos e Técnicas. I.M.Souza Gimenes, E.H.M.Huzita (orgs.) Rio de Janeiro, Ciência Moderna, 2005. p. 57-103.

[24] Wills, A.L. Components and Connectors: Catalysis Techniques for Designing Component Infrastructures, in: Component-Based Software Engineering: Putting the Pieces Together, Hineman, G.T. & Councill, W.T. (eds), Addison-Wesley, 2001.

Um Ambiente para Integração de Aplicações Colaborativas

Roberta Lima Gomes
DI/UFES
Vitória-ES
Brasil
+55 (27) 33 35 26 54
rgomes@inf.ufes.br

Guillermo de J. Hoyos Rivera
Universidad Veracruzana
Vera Cruz
Mexico
+52 (228) 8 17 28 55
ghoyos@uv.mx

Jean Pierre Courtiat
LAAS-CNRS
Toulouse
France
+33 (0) 5 61 33 62 63
courtiat@laas.fr

RESUMO

As atividades colaborativas são geralmente conduzidas por grupos de indivíduos apresentando diferentes requisitos e tarefas a serem executadas. Conseqüentemente, diferentes aplicações colaborativas devem ser utilizadas a fim de realizar essas atividades. No entanto, apesar de serem utilizadas durante a execução de uma mesma atividade colaborativa, as aplicações são executadas de forma independente. A integração de tais aplicações permitiria que suas diferentes funcionalidades de colaboração fossem dinamicamente combinadas e controladas de acordo com as necessidades dos usuários. Este artigo apresenta um ambiente de integração, denominado LEICA, que adota uma abordagem fracamente acoplada permitindo a integração de aplicações colaborativas existentes. Apesar de sua abordagem fracamente acoplada, LEICA oferece mais do que uma simples interoperabilidade entre aplicações, onde o próprio usuário define a semântica de integração desejada.

ABSTRACT

Collaboration activities usually involve several people with different group tasks and needs. Accordingly, different types of collaborative applications are usually applied in order to support group work. But in spite of being used to accomplish a common collaboration task, these applications are executed independently. The integration of such applications would allow their different functionalities to be combined in a controlled way according to user requirements. In order to achieve integration, we propose LEICA, a loosely-coupled integration environment which allows collaborative applications to interact without loosing their autonomy. Despite been loosely-coupled, LEICA goes beyond interoperability, allowing users to define integration semantics.

Palavras-chave

Aplicações colaborativas; CSCW; ambientes de integração; aplicações distribuídas; serviços Web.

Keywords

Collaborative applications; CSCW; integration environments; distributed applications; Web services.

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 68-77.

1. INTRODUÇÃO

Devido ao fato das atividades colaborativas envolverem diferentes indivíduos, apresentando diferentes tarefas de grupo e requisitos, o suporte a essas atividades é dificilmente oferecido por um sistema colaborativo simples. Na verdade, várias funcionalidades de comunicação, coordenação e cooperação podem ser necessárias para a realização de tais atividades. Com isso, alguns sistemas de suporte ao trabalho cooperativo (ou sistema CSCW, *Computer Supported Cooperative Work*) procuram combinar diferentes funcionalidades de forma a oferecer um ambiente único e completo de suporte à colaboração.

O principal problema desta abordagem está associado ao fato de que é praticamente impossível prever todas as funcionalidades necessárias à execução de atividades colaborativas. Devido ao seu aspecto dinâmico, novas exigências podem surgir ao longo de uma atividade. Desta forma, dificilmente um sistema colaborativo será suficientemente apropriado e completo. Com isso, esses sistemas devem ser flexíveis o suficiente para se adaptarem às novas exigências dos usuários [17]. No entanto, os níveis de flexibilidade dos sistemas CSCW atuais são relativamente restritos. Em geral, para se adicionar uma nova funcionalidade colaborativa ao sistema, um novo componente ou módulo deve ser desenvolvido especificamente para este fim.

Consequentemente, para se obter um suporte completo à execução de atividades colaborativas, os usuários decidem utilizar diferentes aplicações colaborativas independentes (acadêmicas ou disponíveis no mercado). Mas apesar de serem utilizadas durante a realização de uma mesma tarefa (ou um conjunto de tarefas), essas aplicações são executadas lado a lado, mas independentemente umas das outras. Um ambiente integrado de colaboração permitiria que as funcionalidades dessas aplicações fossem combinadas e controladas dinamicamente, de acordo com a necessidade dos usuários.

Com o objetivo de possibilitar a integração de diferentes aplicações colaborativas existentes, este trabalho propõe o ambiente LEICA: "*Loosely-coupled Environment for Integrating Collaborative Applications*". LEICA adota uma integração fracamente acoplada que permite a abstração dos detalhes internos de cada aplicação, facilitando assim o processo de integração. Uma outra vantagem desta abordagem é que, uma vez integrada ao ambiente, a aplicação colaborativa mantém sua autonomia de execução. Da mesma forma, LEICA mantém-se independente das aplicações, isto é, novas aplicações podem ser integradas ou retiradas do ambiente sem comprometer a execução deste último.

Para implementar uma integração fracamente acoplada, LEICA define uma arquitetura apoiada em tecnologias de serviços Web e um Sistema de Notificação de Eventos. Uma vez integradas ao ambiente, as aplicações colaborativas passam a interagir dentro do contexto de uma seção colaborativa global, denominada *SuperSession*. Essas interações são controladas segundo a *Política de Colaboração* definida para a respectiva *SuperSession*. Através da Política de Colaboração os usuários especificam como as diferentes aplicações são coordenadas, definindo assim a semântica desejada para a integração.

Este artigo está estruturado em sete seções. A seção 2 apresenta os trabalhos relacionados. Na seção 3 é introduzida a abordagem geral de integração. A seção 4 apresenta o processo de configuração de uma *SuperSession*, detalhando a especificação de sua Política de Colaboração. Na seção 5 é detalhada a arquitetura de LEICA. A seção 6 apresenta a metodologia de análise e projeto do ambiente, finalizando com a descrição do protótipo desenvolvido. Na seção 7 são apontadas algumas conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

Diferentes trabalhos podem ser encontrados na literatura abordando a integração de aplicações heterogêneas, principalmente no contexto de sistemas distribuídos. Soluções genéricas de integração baseadas em *middlewares*, como *CORBA*, *DCOM* e *EJB* encontram-se disponíveis, assim como soluções mais específicas a determinados domínios, como, por exemplo, as soluções de *EAI* (*Enterprise Application Integration*) [1], visando a integração de sistemas de informação empresariais. Recentemente, com a emergência dos serviços Web [20], novas soluções de integração vêm sendo propostas. Devido à utilização de padrões e protocolos abertos, o conceito de serviços Web tem se mostrado como uma tecnologia promissora para a integração de aplicações distribuídas.

As categorias de soluções citadas no parágrafo precedente apresentam na verdade respostas técnicas ao problema da interoperabilidade entre aplicações heterogêneas. Elas oferecem mecanismos permitindo que as aplicações se comuniquem e interajam através da troca de informações. No entanto, essas soluções garantem unicamente a chamada "interoperabilidade sintática" ou "*syntactic interoperability*" [18]. De acordo com Brownsword *et al.* [3] a interoperabilidade deve representar mais do que a capacidade de compartilhar e trocar informações específicas. As aplicações devem estar de acordo quanto ao significado (ou à semântica) dessas trocas, e quanto à maneira de reagir a essas interações. Em outras palavras, deve-se definir uma semântica de integração.

Na área do CSCW, os pesquisadores chegaram à conclusão de que esta interoperabilidade sintática não é suficiente para construir um ambiente de colaboração integrado. Desta forma, alguns trabalhos foram propostos tendo em vista especificamente a integração de aplicações colaborativas. Focalizando-se na integração das funcionalidades de colaboração oferecidas pelas aplicações, estes ambientes buscam a definição de uma semântica por trás desta integração.

Dewan e Sharma [6] abordam a integração de aplicações colaborativas que manipulam de maneira concorrente os mesmos artefatos (por exemplo, dois editores de textos colaborativos distintos manipulando os mesmos documentos textuais). O trabalho se focaliza na integração dos mecanismos de controle de acesso (ou *floor control*) empregados pelas diferentes aplicações acessando um mesmo artefato. Em contrapartida, este trabalho não considera a integração de aplicações que, embora implicadas em uma mesma atividade colaborativa, não manipulem os mesmos artefatos (por exemplo, uma videoconferência e um *Whiteboard* compartilhado).

Iqbal *et al.* [16] propõem um *framework* de integração de sistemas CSCW, apoiando-se nos três modelos apresentados por Ellis e Wainer [8]:

- o modelo de ontologias especifica todos os objetos compartilhados pela aplicação, suas relações e terminologias;
- o modelo de coordenação especifica a maneira em que as interações ocorrem no sistema sob forma de fluxo de processos (ou *workflow*);
- o modelo de interface de usuário descreve como o sistema é apresentado ao usuário final.

O *framework* apresentado em [16] prevê uma integração no nível de cada um desses modelos. O processo de integração consiste em identificar, para cada aplicação, os elementos associados aos três modelos. Em seguida, no nível de cada modelo, os elementos de todas as aplicações são agrupados (onde os elementos equivalentes são "agregados") de forma a gerar modelos comuns de ontologias, coordenação e interface de usuário. Deve-se observar que, apesar de garantir uma integração completa das aplicações colaborativas, esta abordagem requer um bom conhecimento interno de cada aplicação. Consequentemente, a integração de aplicações já existentes, desenvolvidas por terceiros (acadêmicas ou comerciais) torna-se uma tarefa muito complexa (ou mesmo impossível) de ser realizada.

Com o objetivo de evitar que os detalhes internos das aplicações sejam considerados durante o processo de integração (facilitando desta forma a integração de aplicações existentes), alguns ambientes propõem uma abordagem de integração dita "fracamente acoplada", ou "*loosely-coupled*". Esta abordagem apresenta outros dois aspectos importantes:

- uma vez integrada ao ambiente, a aplicação colaborativa preserva sua autonomia de execução, isto é, ela pode continuar a ser utilizada fora do ambiente de integração;
- o ambiente de integração mantém-se independente das aplicações, e consequentemente, aplicações podem ser integradas e retiradas do ambiente sem que o comportamento deste último seja alterado, ou que sua execução seja interrompida.

Esta última característica é particularmente importante se nós considerarmos que um sistema colaborativo deve ser flexível. De fato, em um ambiente fracamente acoplado, o conjunto de aplicações integradas pode ser mais facilmente alterado em função das necessidades dos usuários.

Os sistemas AREA [11] e NESSIE [19] se apóiam em uma abordagem fracamente acoplada para realizar a integração de aplicações colaborativas. Cada um desses sistemas representa um ambiente colaborativo onde diferentes aplicações independentes podem compartilhar um espaço de informações comum, implementado através de uma infra-estrutura de notificação de eventos. O objetivo é oferecer aos usuários das diferentes aplicações uma noção global de presença (ou *awareness*). Os

usuários podem receber notificações de eventos considerados pertinentes à atividade colaborativa global, provenientes de diferentes aplicações (utilizadas por outros membros do grupo). Um aspecto interessante desses dois sistemas é a utilização de tecnologias Internet abertas (como HTTP e CGI) para facilitar a integração de aplicações colaborativas disponíveis no mercado. No entanto, a maior restrição desses dois sistemas é o fato de que a semântica de integração é definida estaticamente – as aplicações colaborativas são integradas com o único objetivo de oferecer uma noção de presença global aos usuários.

Uma outra proposta igualmente baseada em uma abordagem fracamente acoplada é o *framework* XGSP [10]. XGSP prevê a integração de aplicações de videoconferência e audioconferência apoiadas nos padrões SIP, e H.323, assim como as aplicações do *Access Grid* [15]. Neste *framework*, servidores de gerência XGSP (*XGSP collaboration manager servers*) são encarregados de controlar a sessão colaborativa. Para cada tipo de aplicação (isto é, as aplicações SIP, H.323 e *Access Grid*) é definido um *gateway* capaz de se comunicar com servidores XGSP utilizando um protocolo comum de sinalização baseado em serviços Web. A principal limitação de XGSP refere-se ao fato de permitir, originalmente, apenas a integração de aplicações SIP, H.323 e *Access Grid*.

Dustdar *et al.* [7] discutem a importância dos serviços Web para viabilizar a interoperabilidade entre aplicações colaborativas. É proposta uma arquitetura de gerência e configuração de serviços Web apresentando uma camada de serviços para o trabalho em equipe, denominada *teamwork services layer*. O objetivo desta camada é oferecer um conjunto de serviços comuns às aplicações colaborativas (por exemplo, gerência de grupos, gerência de arquivos, agendas compartilhadas, *etc.*). Estes serviços podem ser utilizados de maneira uniforme pelas diferentes aplicações integradas. No entanto, para que as aplicações colaborativas possam ser integradas, os autores partem do princípio de que as mesmas suportam as novas tecnologias de serviços Web. Mesmo que isso represente uma tendência para as novas aplicações colaborativas, atualmente apenas um número limitado de aplicações suporta estas tecnologias.

Vale ressaltar que, na verdade, a utilização de serviços Web também pode trazer outros inconvenientes, particularmente ligados à utilização do protocolo SOAP (*Simple Object Access Protocol* [21]). Este protocolo representa uma camada suplementar, resultando em um atraso adicional no tratamento das mensagens trocadas entre as aplicações [5]. Com isso, Alonso *et al.* [2] sugerem que as tecnologias de serviços Web sejam utilizadas somente para executar operações consideradas como *coarse-grained*, onde o impacto da sobrecarga associada ao tratamento das mensagens SOAP seja menor.

Com o objetivo de tornar possível a integração de aplicações colaborativas existentes, LEICA também define uma abordagem fracamente acoplada de integração apoiada sobre as tecnologias de serviços Web. No entanto, de acordo com as recomendações de Alonso *et al.*, as tecnologias de serviços Web são utilizadas de forma restrita. Como será apresentado a seguir, LEICA propõe uma arquitetura híbrida onde serviços Web são utilizados somente durante as etapas iniciais do processo de integração, assim como durante a criação e o lançamento da execução de sessões colaborativas integradas. Durante a execução de uma sessão colaborativa integrada, as aplicações colaborativas passam a interagir utilizando um outro paradigma de comunicação.

Na próxima seção será apresentada uma visão geral do *framework* definindo a abordagem de integração utilizada por LEICA.

3. ABORDAGEM GERAL DE INTEGRAÇÃO

Antes de descrever o *framework* sobre o qual se apóia LEICA, dois cenários de integração são apresentados a fim de ilustrar o propósito de um tal ambiente de integração.

3.1 Cenários de Integração

a) Ferramenta de navegação Web colaborativa integrada a um Chat

Uma ferramenta de navegação Web colaborativa permite que diferentes usuários naveguem em grupo a Web. A navegação pode ser estruturada estaticamente, seguindo a filosofia "mestre-escravo", ou de forma mais dinâmica, como no caso da aplicação *CoLab* [12]. Quando um usuário se conecta a uma sessão de *CoLab*, ele inicia sua atividade de navegação de forma independente (ou assíncrona). Durante a sessão, um usuário pode decidir se sincronizar com outro usuário, perdendo sua autonomia de navegação, passando a "seguir" a atividade de navegação do outro usuário. À medida que relações de sincronização são estabelecidas entre os usuários, árvores de navegação se formam, onde cada árvore representa um grupo de usuários (cada nó corresponde a um usuário) cuja atividade de navegação é guiada pelo usuário que se encontra na raiz.

Um *Chat* multi-salas poderia ser integrado a *CoLab* definindo-se a seguinte semântica de integração: a cada árvore de navegação de *CoLab* é associada uma sala de discussão do *Chat*, permitindo que os usuários fazendo parte de uma mesma árvore possam discutir sobre a atividade de navegação; desta forma se, por exemplo, um usuário deixar uma árvore de navegação para se conectar a uma outra árvore, ele deve ser automaticamente transferido para a respectiva sala de *Chat*.

b) E-learning

Os sistemas CSCW têm sido amplamente aplicados no contexto de *E-learning*[1]. Desta forma um cenário integrando diferentes aplicações pode ser imaginado.

Um CVE (*Collaborative Virtual Environment*[2]) pode ser utilizado para representar o edifício de uma escola, constituído de: um hall de entrada, duas salas de aula e uma sala de professores. Com a integração de outras aplicações colaborativas, diferentes semânticas de integração podem ser imaginadas. Por exemplo, aplicações colaborativas podem ser associadas às salas do edifício, como um *Chat* associado ao hall de entrada, assim como um *Whiteboard* compartilhado e uma audioconferência, associados a cada uma das salas de aula. Em seguida, pode-se definir que, quando um usuário entra em uma determinada sala, ele seja automaticamente conectado às aplicações colaborativas associadas.

---

[1] O *E-learning* cooperativo deu origem à área de pesquisa intitulada *Computer Supported Cooperative Learning* (CSCL).

[2] Mundo virtual compartilhado geralmente representado em realidade virtual.

Neste cenário, tanto o *Whiteboard* quanto a audioconferência definem um *floor control*: quem detém o *floor* do *Whiteboard* tem o direito de editar/modificar o quadro; quem detém o *floor* da audioconferência é o único a poder falar. Desta forma, uma outra semântica de integração podendo ser definida é a vinculação dos *floors* (ou *floor coupling*) dessas duas aplicações, garantindo que durante uma aula virtual, o usuário que controla o *Whiteboard* (por exemplo, o professor) seja o único que possa falar na audioconferência.

3.2 *Framework* de Integração

A integração de uma aplicação colaborativa a LEICA é realizada associando-se um *Wrapper* a esta aplicação. Para tanto, a aplicação colaborativa deve oferecer uma API (*Application Programming Interface*) dando acesso à mesma. Através desta API o *Wrapper* será capaz de interagir com a aplicação. Portanto o nível de interação dependerá diretamente das funcionalidades oferecidas pela API. Na verdade, acreditamos que os fornecedores/criadores de aplicações colaborativas estejam interessados em criar aplicações que possam ser utilizadas tanto como ferramentas independentes quanto integradas a outras aplicações (através de uma API que seja a mais flexível possível). Nós podemos, por exemplo, citar a ferramenta de comunicação Skype™ que, recentemente, publicou sua API.

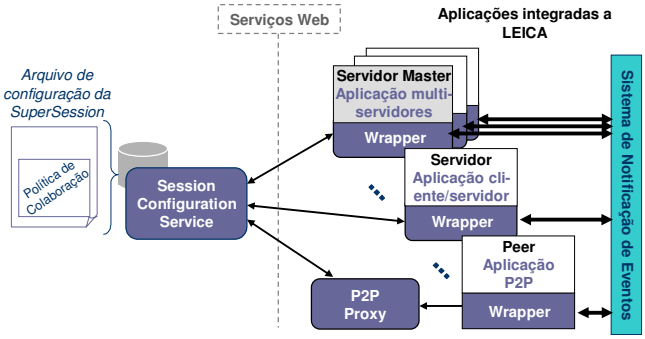


Figura 1. Princípio arquitetural de LEICA.

Como ilustrado na Figura 1, *Wrappers* são associados aos servidores de aplicações cliente/servidor ou multi-servidores, e aos *peers* de aplicações *peer-to-peer* (P2P). Cada *Wrapper* associado a um servidor apresenta uma interface de serviços Web permitindo à respectiva aplicação interagir com LEICA. Mais precisamente, através de sua interface de serviços Web, a aplicação pode se registrar junto ao ambiente e interagir em seguida com o *Session Configuration Service* (ou SCS). Os *Wrappers* associados às aplicações P2P não apresentam uma interface de serviços Web. Essas aplicações devem passar pelo *P2P Proxy* para se registrarem junto a LEICA[1]. Em seguida, o *P2P Proxy* passa a representar essas aplicações interagindo com o SCS.

O SCS é um serviço Web utilizado para (i) configurar novas sessões colaborativas globais, denominadas *SuperSessions*, e (ii) iniciar a execução de *SuperSessions*. Uma *SuperSession* é uma sessão colaborativa integra englobando todos os elementos de uma atividade colaborativa. Diferentes aplicações colaborativas podem então ser utilizadas como "suporte" de uma *SuperSession*, onde cada aplicação gerencia suas próprias sessões colaborativas, denominadas *specificSessions* (por exemplo, sessões de videoconferência ou de *Whiteboard* compartilhado).

Durante o processo de configuração de uma *SuperSession*, o SCS entra em contato com o *Wrapper* de cada aplicação colaborativa requisitando: (i) quais são os tipos de dados necessários à criação de uma nova *specificSession* para esta aplicação (por exemplo, uma ferramenta de videoconferência pode informar que ela precisa de um endereço IP *multicast* para configurar uma sessão); e (ii) quais são os tipos de eventos que ela é capaz de notificar, assim como os tipos de requisições de ação que ela é capaz de tratar[2]. Este segundo conjunto de informações será particularmente necessário durante a especificação da Política de Colaboração da *SuperSession*.

Durante a configuração de uma *SuperSession*, além de especificar as informações gerais de gerência, e de escolher e configurar as aplicações a serem utilizadas como suporte da *SuperSession* (dentro do conjunto de aplicações integradas a LEICA), o usuário deve especificar a Política de Colaboração desejada para esta *SuperSession*. Esta Política de Colaboração é constituída de um conjunto de regras seguindo um modelo eventos-condições-ações. Essas regras definem como cada aplicação colaborativa deve reagir quando eventos são notificados por outras aplicações. Em outras palavras, as regras determinam as ações que as aplicações (participando de uma *SuperSession*) devem executar em resposta às notificações de eventos recebidas. Com isso, a Política de Colaboração define como as diferentes aplicações são coordenadas durante a execução de uma *SuperSession*, oferecendo aos usuários a flexibilidade de definir a semântica de integração de acordo com as suas necessidades (sem impor esta semântica).

Uma vez que uma *SuperSession* foi configurada (e o arquivo de configuração correspondente foi gerado), o SCS pode finalmente iniciar sua execução. Inicialmente, ele entra em contato com cada aplicação colaborativa escolhida para informá-la de que a mesma será utilizada durante esta *SuperSession*, repassando as respectivas informações de configuração (como por exemplo, as informações para a criação de *specificSessions*). Em seguida os *Wrappers* das aplicações se interconectam através do Sistema de Notificação de Eventos (a partir deste ponto, as interfaces de serviços Web não são mais utilizadas).

Com o decorrer da atividade colaborativa, os *Wrappers* das aplicações utilizadas trocam notificações de eventos em modo P2P. Os *Wrappers* são igualmente responsáveis pela execução da Política de Colaboração. Assim, sempre que um *Wrapper* receber uma notificação de evento, ele verifica se este evento sensibiliza alguma regra relativa à aplicação à qual ele encontra-se associado (isto é, uma regra apresentando uma requisição de ação destinada a esta aplicação). Se uma sensibilização ocorre, ele envia a requisição de ação à respectiva aplicação, para que a mesma a execute.

LEICA não tem como objetivo ser capaz de considerar a notificação de eventos físicos de baixo nível (como por exemplo,

[1] Visto que geralmente as aplicações P2P não permanecem constantemente disponíveis nos computadores dos usuários, foi decidido que as mesmas não serão expostas como serviços Web.

[2] Os tipos de eventos e ações estão diretamente relacionados com a API disponibilizada pela aplicação.

clique ou movimentos de mouse) nem de eventos de sincronização apresentando uma frequência elevada (como por exemplo, a posição atual de um objeto em deslocamento). Na verdade, LEICA pressupõe que as aplicações colaborativas sejam capazes de notificar eventos veiculando uma "semântica aplicativa", isto é, eventos que possam ser relevantes para a atividade colaborativa. Fazendo referência ao cenário de *E-learning* descrito da seção 3.1, exemplos de eventos seriam: o fato de um usuário entrar em uma sala virtual do CVE; a passagem do *floor* do *Whiteboard* de um usuário a outro.

Na próxima seção, o processo de configuração de uma *SuperSession* é detalhado, explicando-se como uma Política de Colaboração deve ser especificada.

4. CONFIGURAÇÃO DE UMA *SUPERSESSION*

A configuração de uma *SuperSession* é realizada em duas etapas: (i) configuração das informações de gerência e (ii) configuração da Política de Colaboração.

4.1 Configuração das Informações de Gerência

As informações de gerência a serem configuradas se dividem em dois grupos:

- GSMinfo (*General Session Management information*) – este grupo corresponde às informações de gerência gerais da *SuperSession*, tais como nome, propósito, *scheduling*, os possíveis participantes (*membership*), papéis de usuários (ou *roles*), *etc*.
- IAinfo (*Integrated Applications information*) – para cada aplicação a ser utilizada como suporte da *SuperSession*, é configurada a lista de *specificSessions*, onde informações específicas a cada aplicação são fornecidas (por exemplo, para se configurar uma sessão de videoconferência, será fornecido um endereço IP *multicast*).

4.2 Configuração da Política de Colaboração

A associação de uma Política de Colaboração a uma *SuperSession* representa um elemento chave do ambiente. É a especificação dessa política que permite ao usuário definir a semântica de integração, especificando-se o comportamento desejado para as aplicações colaborativas durante a execução de uma *SuperSession*.

A Política de Colaboração corresponde a um conjunto de regras de colaboração, onde cada regra permite associar notificações de eventos a requisições de execução de ação, sob determinadas condições. Portanto, através dessas regras é possível vincular as atividades sendo realizadas em diferentes aplicações colaborativas.

Procurou-se definir um mecanismo para a especificação de regras que fosse o mais intuitivo possível para os usuários, e ao mesmo tempo oferecesse um modelo mais elaborado que o clássico "evento-condição-ação". Desta forma foi escolhido um método gráfico para a especificação das regras de colaboração. Por meio simples de um editor gráfico, o usuário pode definir regras compondo elementos gráficos denominados *policy widgets*. Através destes *widgets* é possível definir regras simples, associando uma única notificação de evento a uma única requisição de ação, ou regras mais complexas associando *n* notificação de eventos à *m* requisições de ação.

Para compor estas regras, é necessário conhecer *a priori* os tipos de eventos que cada aplicação integrada a LEICA é capaz de notificar, assim como os tipos de requisições de ação que cada aplicação é capaz de tratar. É importante ressaltar que cada aplicação pode definir os tipos de eventos e ações que for necessário. Os possíveis conflitos na "tipagem" desses eventos/ações (por exemplo, duas aplicações especificam um mesmo tipo de evento[1]) são resolvidos associando-se o identificador único da aplicação a cada tipo de evento ou ação.

A partir da especificação gráfica das regras compondo a Política de Colaboração, é gerada uma especificação textual correspondente que é então anexada ao arquivo de configuração da respectiva *SuperSession*. A semântica dessas regras de colaboração é definida por tradução em Redes de Petri. Esta semântica é implementada por um sub-módulo do *Wrapper* responsável pela execução das Políticas de Colaboração.

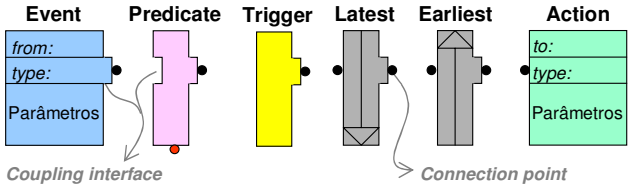


Figura 2. Policy *Widgets*.

A Figura 2 ilustra os seis *widgets* utilizados para a definição das regras de colaboração. Os *widgets* podem ser conectados através de seus respectivos "*connection points*" ou acoplados através de suas "*coupling interface*". Cada regra é então o resultado da composição desses diferentes *widgets* onde o princípio básico de composição é o seguinte:

(i) as regras são lidas da esquerda para a direita;
(ii) somente *widgets* que não apresentem *connection points* nem *coupling interfaces* do lado esquerdo (*i.e. Event* e *Trigger*) podem aparecer na extremidade esquerda de uma regra;
(iii) somente *widgets* que não apresentem *connection points* nem *coupling interfaces* do lado direito (*i.e. Action*) podem aparecer na extremidade direita de uma regra;
(iv) cada regra deve ser composta por pelo menos um *widget Event* (ou *Trigger*) e um *widget Action*.

Durante a execução de uma *SuperSession*, as regras definindo a sua Política de Colaboração são executadas em paralelo. Uma regra em execução é dita ativa. Uma regra é sensibilizada quando todos os seus componentes são sensibilizados.

O *widget Event* representa uma notificação de evento. Cada *Event* é associado a uma aplicação colaborativa e é caracterizado por um tipo (através dos campos "*from*" e "*type*" respectivamente). Ele apresenta uma lista de parâmetros cujos valores podem ser filtrados (usando-se *matching patterns*). O *widget Action* representa uma requisição de execução de ação. Cada *Action* é destinado a uma aplicação colaborativa (através do campo "*to*") e

[1] Mesmo que duas ou mais aplicações especifiquem um mesmo tipo de evento, a semântica associada ao evento será específica a cada aplicação.

é caracterizado por um tipo (através do campo "*type*"). Este *widget* apresenta uma lista de parâmetros cujos valores devem ser especificados. Vale ressaltar que todos os tipos possíveis de *Events* e *Actions* (assim como seus parâmetros) são conhecidos, informação esta fornecida pelas próprias aplicações colaborativas[1].

A Figura 3 ilustra uma regra simples que associa diretamente um *Event* a um *Action*. Esta regra é sensibilizada assim que o evento especificado for notificado. Ela define que: se a aplicação "CA1" notificar um evento de tipo "T1" apresentando os parâmetros "a" com um valor "M*" (uma string cujo primeiro caractere é 'M') e "b" com o valor ">10", então deve ser enviada à aplicação "CA2" uma requisição para que a mesma execute a ação de tipo "T2". O caracter '%' é um operador de referência, indicando que o parâmetro "y" deve ter seu valor copiado do parâmetro "a" presente no *widget Event* que apresenta o identificador "1"[2]. Parâmetros também podem ter seus valores determinados a partir do estado atual da *SuperSession* (como no caso do parâmetro "z" que recebe como valor o número de usuários conectados na *SuperSession* no instante em que a regra é sensibilizada).

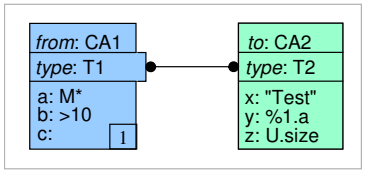


Figura 3. Exemplo de regra "evento-ação".

Através dos *widgets Predicate* e *Trigger* é possível impor condições (sob a forma de predicados) para que uma regra seja sensibilizada.

O *Predicate* pode ser associado (acoplado) a todos os outros *widgets*, com exceção do *Action*. O predicado definido deve ser avaliado sempre que o *widget* associado for sensibilizado. Em geral, este predicado pode depender de variáveis do estado global da *SuperSession* e de parâmetros de *widgets Event* presentes na regra. O quadro (a) da Figura 4 ilustra um exemplo de regra apresentando um *widget Predicate*. Esta regra especifica que sempre que o evento "T1" for notificado por "CA1", se o predicado for verdadeiro (*i.e.* se o número de usuários conectados for superior ou igual a 10) uma requisição de ação de tipo "T3" deve ser enviada a "CA2" (como é o caso no instante t_3). Uma particularidade do *Predicate* é apresentada neste exemplo, onde um comportamento alternativo (através do *connection point* posicionado embaixo do *widget*) pode ser definido quando o mesmo for avaliado como falso[3]. Assim, no instante t_1, a regra é igualmente sensibilizada, mas neste caso uma requisição de ação de tipo "T2" deve ser envidada a "CA2".

Ao contrário do *Predicate*, o predicado especificado pelo *widget Trigger* é avaliado de forma contínua, e cada vez que ele se tornar verdadeiro, o *widget* é sensibilizado (como é o caso nos instantes t_1 et t_3 no quadro (b) da Figura 4). Uma outra diferença é que ele só pode impor condições sobre o estado atual da *SuperSession*. Desta forma, o *Trigger* apresenta uma semântica similar à de um evento, "notificando" modificações no estado da *SuperSession*.

Através dos *widgets Earliest* e *Latest* é possível compor diferentes notificações de eventos dentro de uma mesma regra. Quando *Events* são agrupados através de um *Earliest*, a regra é sensibilizada assim que um dos eventos especificados for notificado. Quando *Events* são agrupados através de um *Latest*, a regra é sensibilizada apenas após a notificação de todos os eventos especificados. A Figura 5 ilustra dois exemplos de regras utilizando *Earliest* (a) e *Latest* (b), ambos associados a um *Predicate*. Visto que o *Earliest* define um comportamento não-determinístico (não se sabe previamente qual dos eventos especificados será notificado), apenas no caso do *Latest* o predicado poderá referenciar parâmetros dos eventos notificados.

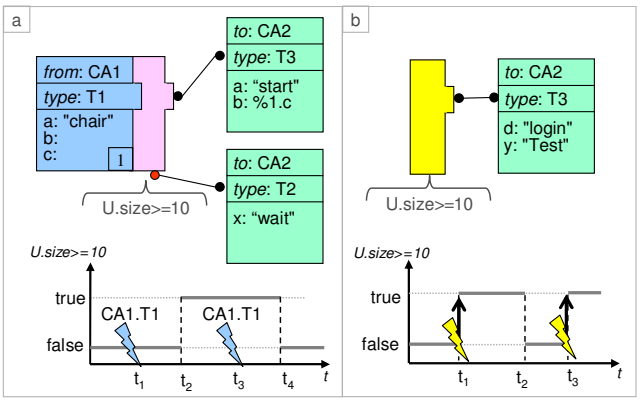


Figura 4. Exemplo de regras apresentando predicados.

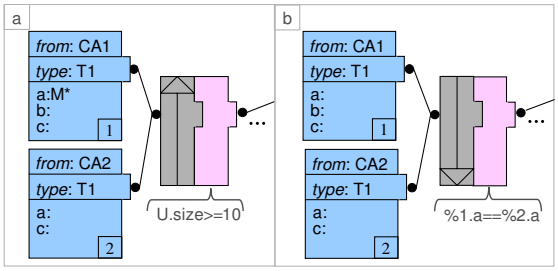


Figura 5. Exemplo de regras utilizando *Earliest* e *Latest*.

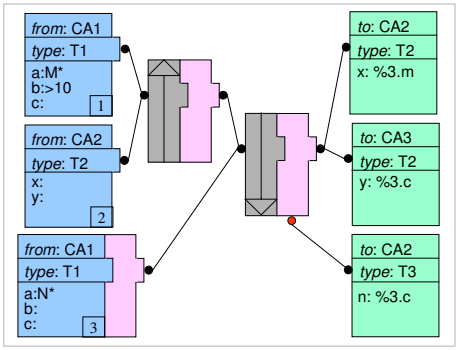


Figura 6. Exemplo de regra complexa.

---

[1] Existem também tipos de eventos e ações predefinidos pelo próprio ambiente, como CONNECT e DISCONNECT.

[2] Todo *Event* é automaticamente associado a um identificador numérico.

[3] Um comportamento alternativo só pode ser definido quando à direita do *Predicate* só houver *widgets Action*.

Como ilustrado na Figura 6, diferentes *Earliests* e *Latests* podem ser combinados a fim de compor regras mais complexas.

5. ARQUITETURA: DA INTEGRAÇÃO DE APLICAÇÕES À EXECUÇÃO DE SUPERSESSIONS

De acordo com o *framework* geral de integração apresentado na seção 3, três etapas principais são necessárias para implementar as *SuperSessions*: (i) a integração de aplicações colaborativas a LEICA, (ii) a criação de novas *SuperSessions*, e, finalmente , (iii) a execução das *SuperSessions*.

Para melhor ilustrar a arquitetura global do sistema, identificando precisamente o papel de cada módulo e suas interações, a arquitetura será apresentada percorrendo-se cada uma das etapas do *framework* de integração.

5.1 Integração de uma Aplicação Colaborativa a LEICA

Ao integrar uma aplicação apresentando uma arquitetura cliente/servidor ou multi-servidores a LEICA, um *Server Wrapper* deve ser associado ao respectivo servidor (ou servidores). Já no caso das aplicações distribuídas segundo o paradigma P2P, um *P2P Wrapper* é associado a cada aplicação *peer*. Como ilustrado na Figura 7, a diferença básica entre esses dois *Wrappers* encontra-se na interface de serviços Web (*WS Interface*), presente apenas no *Server Wrapper*. O *P2P Wrapper* apresenta no lugar uma interface específica para a comunicação com o *P2P Proxy*.

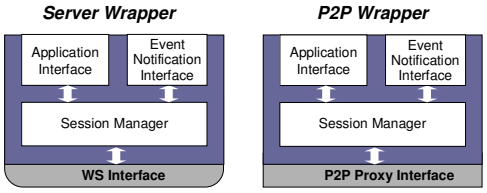


Figura 7. Os dois tipos de *Wrappers* de LEICA.

Com o objetivo de diminuir os esforços de implementação durante a integração de uma aplicação a LEICA, foi definido um módulo especial, denominado *API Factory*. Este módulo, baseado em uma descrição XML[1] da API de uma aplicação, gera automaticamente um *Wrapper* adaptado a esta aplicação. Esta adaptação consiste basicamente em criar (de forma dinâmica) o sub-módulo *Application Interface*, definindo dentro deste último um conjunto de métodos de acordo com a API da aplicação.

Application Interface é o sub-módulo do *Wrapper* que deve ser diretamente interligado à aplicação colaborativa. Através dele, a aplicação poderá (i) notificar ao *Wrapper* "o que está acontecendo" dentro de seu contexto de colaboração, e (ii) receber requisições de execução de ação (como a criação de uma nova *specificSession*).

[1] Para cada aplicação colaborativa, dois arquivos devem ser especificados: *Specific Data File* (um *XML Schema*), definindo os tipos de dados que a aplicação necessita para criar *specificSessions*; e *Attributes and API File*, definindo basicamente os tipos de eventos e ações que a aplicação é capaz de tratar.

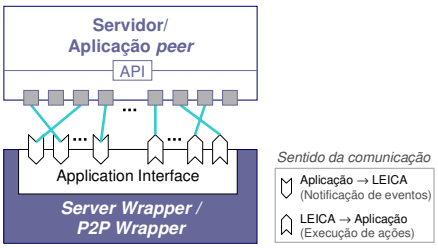


Figura 8. Processo de integração do *Wrapper* à aplicação.

Uma vez gerado o *Wrapper*, o mesmo deve ser integrado à aplicação (servidor ou *peer*). Este processo é realizado manualmente, onde os diferentes métodos da API da aplicação devem ser associados aos métodos gerados no sub-módulo *Aplication Interface* do *Wrapper*, como ilustrado na Figura 8.

Os demais sub-módulos do *Wrapper* são gerados estaticamente (sendo idênticos p/ todos os *Server Wrappers* ou *P2P Wrappers*). O *Event Notification Interface* implementa a interface de comunicação entre o *Wrapper* e o Sistema de Notificação de Eventos. O *Session Manager* implementa as funcionalidades principais do *Wrapper*: (i) receber e tratar os dados de configuração das *SuperSessions*; (ii) decidir quando enviar notificações de evento; (iii) gerenciar a execução das Políticas de Colaboração; (iv) gerenciar o estado atual de cada *SuperSession*.

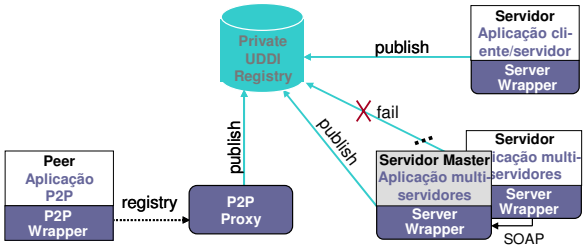


Figura 9. Registro de aplicações colaborativas.

Uma vez que o *Wrapper* foi associado à aplicação colaborativa, a mesma deve se registrar junto a LEICA. Para isso, como ilustrado na Figura 9, o *Wrapper* publica seus serviços em um *UDDI Registry* privado. No caso de aplicações multi-servidores, apenas o primeiro *Wrapper* a ser executado realizará com sucesso esta operação. Ele é então designado como *Master*, enquanto os *Wrappers* dos demais servidores deverão constatá-lo diretamente. No caso de aplicações P2P, este registro é realizado através do *P2P Proxy*.

5.2 Criação de uma *SuperSession*

O *Session Configuration Service* é o módulo responsável por gerenciar a criação de novas *SuperSessions*. Ele representa um serviço Web acessível aos usuários através de uma aplicação Web. Como ilustrado na etapa (1) da Figura 10, quando esta aplicação é acessada, ela entra em contato com o SCS para dar início ao processo de configuração de uma *SuperSession*.

Em seguida, na etapa (2), enquanto o usuário configura as informações de gestão geral (*GSMinfo*), o SCS realiza uma pesquisa no *UDDI Registy* para descobrir quais aplicações colaborativas foram integradas a LEICA. Para cada aplicação, o SCS recebe a URL associada à interface de serviços Web do respectivo *Wrapper*.

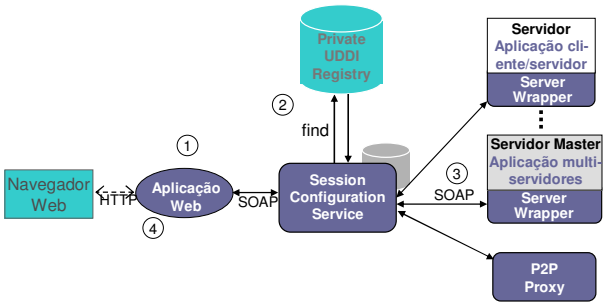


Figura 10. Etapas executadas durante a criação de uma *SuperSession*.

Na etapa (3) o SCS entra em contato com cada uma das aplicações integradas, e com *P2P Proxy*, requisitando: (i) os tipos de dados que cada aplicação necessita para criar novas *specificSessions*, e (ii) os tipos de eventos e ações que cada uma é capaz de tratar. Baseando-se nestas informações, durante a etapa (4) o usuário pode escolher e configurar (especificando o *IAinfo*) as aplicações a serem usadas como suporte da *SuperSession*. Em seguida, ele especifica a Política de Colaboração da *SuperSession*. Finalmente a aplicação Web gera um arquivo de configuração que é enviado ao SCS para que o mesmo o armazene em um banco de dados local.

5.3 Execução de uma *SuperSession*

O SCS é igualmente responsável por dar início à execução de uma *SuperSession*. A execução de uma *SuperSession* pode ser iniciada automaticamente (de acordo com o *scheduling* definido) ou manualmente, como esquematizado na Figura 11:

(1) O usuário acessa a aplicação Web para iniciar a execução de uma *SuperSession*;
(2) O SCS acessa e interpreta o arquivo de configuração da *SuperSession* a fim de identificar quais aplicações colaborativas devem ser utilizadas;
(3) O SCS entra em contato com os *Wrappers* das aplicações para informá-los de que elas farão parte de uma *SuperSession*, repassando a elas as respectivas informações de configuração [1];
(4) Finalmente, os *Server Wrappers* iniciam a execução da Política de Colaboração da *SuperSession* e se interconectam através do Sistema de Notificação de Eventos (não havendo mais uso das interfaces de serviços Web).

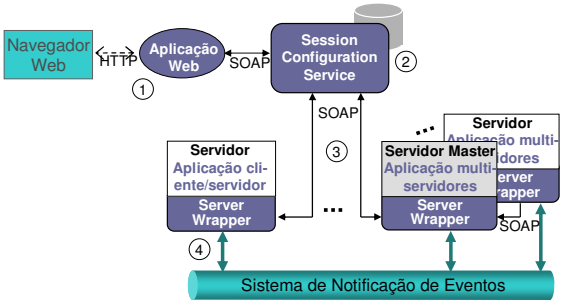


Figura 11. Inicio da execução de uma *SuperSession*.

A execução da Política de Colaboração de uma *SuperSession* é realizada de forma distribuída. Na verdade, cada *Wrapper* só deve executar o subconjunto de regras que definem requisições de ação destinadas à aplicação à qual ele está associado. As regras da Política de Colaboração também orientam os *Wrappers* durante a notificação de eventos. Isto é, somente os eventos que podem sensibilizar alguma regra de colaboração devem ser notificados pelos *Wrappers*[2].

O Sistema de Notificação de Eventos definido por LEICA se baseia no paradigma *publish/subscribe* [9], que define um modelo de comunicação bastante adaptado a ambientes fracamente acoplados. Em geral os assinantes (ou *subscribers*) registram o interesse em receber mensagens associadas a um determinado tópico (ou assunto), mensagens estas enviadas por um conjunto de publicadores (ou *publishers*). Este modelo oferece um desacoplamento entre publicadores e assinantes visto que um não está ciente da presença do outro.

Uma vez que a *SuperSession* tenha entrado em execução, os usuários podem finalmente se conectar ao ambiente. Para tanto, eles devem executar a aplicação *LClient*. A Figura 12 ilustra as etapas executadas durante a conexão de um usuário:

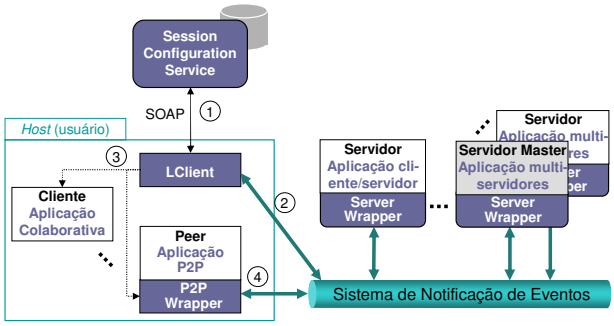


Figura 12. Etapas executadas durante a conexão de um usuário a uma *SuperSession*.

(1) *LClient* solicita ao SCS as informações de configuração da *SuperSession* selecionada pelo usuário;
(2) Assim como os *Wrappers*, o *LClient* executa a Política de Colaboração da *SuperSession*, conectando-se em seguida ao Sistema de Notificação de Eventos.
(3) Sempre que uma regra contendo uma ação de tipo CONNECT for sensibilizada, o *LClient* se encarrega de conectar o usuário executando localmente a respectiva aplicação colaborativa, podendo ser um cliente ou uma aplicação P2P; no segundo caso, ele passa ao *P2P Wrapper* desta aplicação as informações de configuração da *SuperSession*;
(4) O *P2P Wrapper* por sua vez executa a Política de Colaboração e se conecta ao Sistema de Notificação de Eventos.

À medida que os usuários realizam as atividades colaborativas, utilizando as diferentes aplicações no contexto de uma *SuperSession*, os *Wrappers* destas aplicações notificam os eventos relacionados a essas atividades. Em paralelo, a Política de Colaboração é executada por cada *Wrapper* e, em função das notificações de eventos recebidas (e eventualmente de mudanças no estado da *SuperSession*), certas regras podem ser sensibilizadas

[1] Apenas os *Server Wrappers* são contatados inicialmente.

[2] Além dos eventos de tipo CONNECT e DISCONNECT.

resultando na execução de ações por parte dessas aplicações (solicitadas através dos *Wrappers*).

6. ANÁLISE, PROJETO E IMPLEMENTAÇÃO DO AMBIENTE

Com o objetivo de especificar e mesmo formalizar a estrutura e o comportamento dos elementos arquiteturais de LEICA, foi realizada uma modelagem em UML 2.0 (*Unified Modeling Language*) do sistema utilizando-se a ferramenta *TAU G2* de *TeleLogic*.

Foi aplicada uma abordagem *top-down* durante a análise e o projeto do ambiente – partiu-se de uma especificação de "nível 0" sobre a qual foi realizada uma decomposição hierárquica através de refinamentos sucessivos dos cenários (diagramas de caso de uso e de sequência) e arquiteturais (diagramas de classe e de estrutura composta), chegando-se até o "nível 3". Neste último nível, foram definidos os comportamentos dos diferentes componentes e sub-componentes do sistema através de diagramas de estado especificados em SDL (*Specification Description Language*).

Com a definição desses comportamentos, o modelo torna-se "executável", permitindo assim a execução de simulações utilizando-se a ferramenta *Tau G2*. Como resultado das simulações, são gerados *traces* sob a forma de diagramas de sequência, que podem ser então comparados aos diagramas de sequência definidos inicialmente (durante a fase de análise). Devido à complexidade de LEICA, foram realizadas simulações explorando de forma aleatória os diferentes diagramas de estado (ao invés de simulações exaustivas). Com isso foi aplicada uma metodologia de validação informal usando-se técnicas de observação. Procurou-se desta forma tornar a concepção do ambiente confiável e assegurar a consistência entre a implementação final e as especificações iniciais.

Com base no modelo UML especificado, foi desenvolvido um protótipo de LEICA implementando os principais componentes do ambiente[1]. Por questões de portabilidade, Java™ foi escolhida como plataforma de desenvolvimento. Algumas tecnologias Java livres e/ou de código aberto foram igualmente utilizadas:

- Apache jUDDI – implementação do *UDDI Registry*;
- UDDI4J de IBM – implementação das interações entre o *WS Interface* do *Wrapper* e o *UDDI Registry*;
- Apache Tomcat 5.0 et Apache SOAP 2.3 – implementação das interações entre o SCS e os *Wrappers* (e os *LClients*);
- *Scribe* [4] – implementação do Sistema de Notificação de Eventos; *Scribe* implementa o paradigma *publish/subscribe* baseando-se em um mecanismo de multicast aplicativo eficaz.

A fim de testar as funcionalidades implementadas no protótipo, duas aplicações colaborativas (ambas cliente/servidor) foram integradas: (i) *CoLab* [12], uma ferramenta de navegação web colaborativa, e *Babylon Chat* [14], um *Chat* multi-salas. A integração de *CoLab* foi direta uma vez que o mesmo oferece uma API relativamente simples. No caso de *Babylon*, que trata-se de uma aplicação de código aberto, foi necessário desenvolver uma API para poder integrá-la a LEICA.

[1] Não foram implementados os módulos envolvidos na integração de aplicações P2P, nem a aplicação Web de acesso ao SCS.

Após a integração das aplicações ao protótipo, foi criada uma *SuperSession* ("CoLab+Chat") cuja Política de Colaboração define a mesma semântica de integração descrita anteriormente no cenário "a" da seção 3.1. A Figura 13 ilustra o comportamento desta *SuperSession*: o usuário "D" decide se sincronizar como o usuário "F" passando a fazer parte da mesma árvore de navegação que "F" dentro de *CoLab*; com isso, em resposta à execução da Política de Colaboração definida, "D" e todos os outros usuários que pertenciam à árvore de navegação de "D" (neste caso "B") são automaticamente transferidos para a mesma sala de *Chat* de "F".

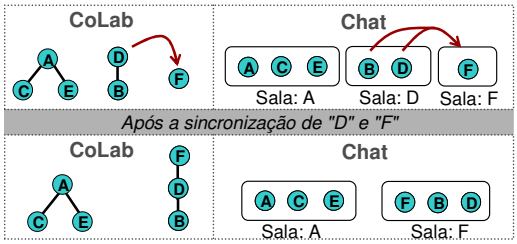


Figura 13. Comportamento da *SuperSession* "CoLab+Chat".

6.1 Considerações sobre o protótipo

Apesar de representar uma implementação parcial do ambiente, o protótipo desenvolvido permitiu a concretização dos principais aspectos da abordagem de integração de LEICA:

a) Integração fracamente acoplada – Esta abordagem contribui significativamente para a simplicidade e a viabilidade do processo de integração, sobretudo por permitir uma integração independente para cada aplicação. Uma desvantagem desta abordagem está relacionada com a implantação do ambiente nos *hosts* dos usuários, onde os clientes (ou os *peers*) das aplicações colaborativas devem ser previamente instalados. Uma solução poderia ser o uso de *Java Web Start* [13] para permitir uma implantação dinâmica das aplicações, no entanto isto restringiria o conjunto de aplicações integradas à tecnologia Java.

b) Uso de serviços Web – Esta tecnologia mostrou-se bastante adaptada para a implementação de uma comunicação fracamente acoplada baseada em mensagens. No entanto, foi verificado, em termos de tempo de resposta, o custo inerente ao emprego de tal tecnologia. Foram realizadas algumas medidas para se calcular aproximadamente o tempo adicionado pela camada de serviços Web. Por exemplo, o tratamento de mensagens SOAP referente aos métodos da interface do SCS é realizado em média em 35ms, enquanto que simulando a transmissão de estas mensagens através de *sockets* TCP simples, o tempo de tratamento reduz-se a 10ms.

c) Sistema de Notificação de Eventos – A utilização de *Scribe* foi de grande importância, sobretudo devido ao seu desempenho. A título ilustrativo, foi conduzido um experimento em uma LAN com cinco nós (simulando 5 *Wrappers*) interconectados através de *Scribe*. Cada nó enviava periodicamente notificações de eventos, dos quais todos os outros nós eram assinantes. A latência média entre o envio e a recepção foi de 300ms. Testes em grande escala foram realizados por Castro *et al.* [4], onde pode-se constatar um desempenho satisfatório de *Scribe* (este último mostrou-se apenas entre 1.59 e 2.2 vezes mais lento do que *IP multicast*).

7. CONCLUSÕES

Neste artigo foi apresentado LEICA, um ambiente fracamente acoplado para a integração de aplicações colaborativas. O ambiente se apóia em tecnologias de serviços Web e um Sistema de Notificação de Eventos para implementar a abordagem de integração fracamente acoplada. No contexto de uma *SuperSession*, uma atividade colaborativa global é realizada usando-se diferentes aplicações colaborativas integradas ao ambiente.

Uma característica chave de LEICA é o fato de que os próprios usuários podem definir a semântica de integração desejada, em função de suas necessidades. Esta semântica é definida através da Política de Colaboração da *SuperSession*, onde um conjunto de regras permitem coordenar as interações entre as diferentes aplicações.

O desenvolvimento do protótipo de LEICA confirmou as vantagens da abordagem fracamente acoplada, que facilitou o processo de integração e garantiu uma autonomia às aplicações integradas. No entanto, a flexibilidade oferecida ao usuário quanto aos tipos de semânticas podendo ser definidas para a integração mostrou uma forte dependência das APIs de cada aplicação.

Como trabalhos futuros, novas aplicações devem ser integradas ao protótipo, como Skype™ e Asterisk (já iniciada). Além disso, testes de desempenho devem ser executados para verificar o comportamento do ambiente em larga escala.

Com relação aos serviços Web, vários esforços vêm sendo dedicados buscando-se soluções para o problema de desempenho de SOAP. Com isso, o Sistema de Notificação de Eventos de LEICA poderia ser igualmente implementado usando-se esta tecnologia, tornando o ambiente completamente aberto.

Com relação às Políticas de Colaboração, uma vez que a semântica das regras foi definida usando-se Redes de Petri, uma verificação baseada neste formalismo poderia ser realizada. Com isso poderia-se, por exemplo, identificar dentro de uma regra a existência de possíveis conflitos ou incoerências.

8. AGRADECIMENTOS

CNPq (Bolsa GDE, Roberta Lima Gomes).
CONACyT/PROMEP (Bolsa 70360, Guillermo J. Hoyos Rivera).

9. REFERÊNCIAS

[1] Alonso, G., Casati, F., Kuno, H., Machiraju, V. "Enterprise Application Integration" In: Web Services – Concepts, Architectures and Applications, Springer pp.67-92, 2004.

[2] Alonso, G., Casati, F., Kuno, H., Machiraju, V. "Web Services" In: Web Services – Concepts, Architectures and Applications, Springer-Verlag pp.123-149, 2004.

[3] Brownsword, L.L. et. al. "Current Perspectives on Interoperability" Technical Report CMU/SEI-2004-TR-009 Software Engineering Institute, Carnegie Mellon University 51pp., Pittsburgh (USA), Mar. 2004.

[4] Castro, M., Druschel, P., Kermarrec, A.-M., Rowstron, A. "Scribe: A large-scale and decentralized application-level multicast infrastructure" IEEE Journal on Selected Areas in Communications (JSAC), v.20, n.8, pp.1489-1499, 2002.

[5] Chiu, K., Govindaraju, M., Bramley, R. "Investigating the limits of SOAP performance for scientific computing" 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11 2002), IEEE Computer Society Press pp.246-254, Edinburgh (Scotland), 2002.

[6] Dewan, P., Sharma, A. "An experiment in interoperating heterogeneous collaborative systems" 6th European Conference on Computer Supported Cooperative Work ECSCW'99), pp.371-390, Copenhagen (Denmark), 1999.

[7] Dustdar, S., Gall, H., Schmidt, R. "Web services for Groupware in Distributed and Mobile Collaboration" 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP'04) pp.241-247, Coruna (Spain), 2004.

[8] Ellis, C.A., and Wainer, J. "A Conceptual Model of Groupware" ACM Conference on Computer Supported Cooperative Work (CSCW'94), ACM press pp.79-88, Chapel Hill (USA), 1994.

[9] Eugster, P., Felber, P., Guerraoui, R., Kermarrec, A.-M. "The many faces of publish/subscribe" ACM Computing Surveys, ACM press v.35, n.2, pp.114-131, 2003.

[10] Fox, G., Wu, W., Uyar, A., Bulut, H., Pallickara, S. "A Web services framework for collaboration and videoconferencing" Workshop on Advanced Collaborative Environments (WACE'03) Seattle (USA), Jun. 2003.

[11] Fuchs, L. "AREA: a cross-application notification service for groupware" 6th European Conference on Computer Supported Coop-erative Work (ECSCW'99) pp.61-80, Copenhagen (Denmark), 1999.

[12] Hoyos-Rivera, G.J.H., Gomes, R.L.; Courtiat, J.P.; Willrich, R.. "CoLab A New Paradigm and Tool for Browsing Collaboratively the Web". IEEE Transactions on Systems, Man and Cybernetics. Part A, Systems and Humans, v.36, n.6, Nov. 2006.

[13] http://java.sun.com/products/javawebstart/

[14] http://visopsys.org/andy/babylon/index.html

[15] http://www.accessgrid.org/

[16] Iqbal, R., James, A., Gatward, R. "A practical solution to the integration of collaborative applications in academic environment" 5th International Workshop on Collaborative Editing Systems, In ECSCW'03 Helsinki (Finland), Sep. 2003.

[17] Kahler, H., Mørch, A., Stiemerling, O., Wulf, V. "Introduction to the Special Issue on Special Issue on Tailorable Systems and Cooperative Work" Computer Supported Cooperative Work: The Journal of Collaborative Computing, v.9, n.1. pp.1-4, 2000.

[18] Lewis, G., Wrage, L. "Approaches to Constructive Interoperability" Technical Report CMU/SEI-2004-TR-020 Software Engineering Institute, Carnegie Mellon University 59pp., Pittsburgh (USA), Dec. 2004.

[19] Prinz, W. "NESSIE: an awareness environment for cooperative settings" 6th European Conference on Computer Supported Cooperative Work (ECSCW'99), Kluwer Academic Publishers pp.391-410, Copenhagen (Denmark), 1999.

[20] W3C – "Requirements for the Internationalization of Web Services" W3C Working Group Note 16 November 2004 http://www.w3.org/TR/ws-i18n-req/

[21] W3C – "SOAP Version 1.2 Part 1: Messaging Framework" W3C Recommendation 24 June 2003.

Gestão de Conhecimento sobre Avaliações de Sistemas Colaborativos

Renata Mendes de Araujo
UNIRIO
Av Pasteur 458, Urca
RJ 22290-240
(+55)(21)25308088
renata.araujo@uniriotec.br

Flavia Maria Santoro
UNIRIO
Av Pasteur 458, Urca
RJ 22290-240
(+55)(21)25308088
flavia.santoro@uniriotec.br

Fernando Prata
DCC-UFRJ
Caixa Postal 68.530
CEP 21941-590
Rio de Janeiro - RJ
(+55)(21)22908091
frprata@gmail.com

Thalita Moraes
DCC-UFRJ
Caixa Postal 68.530
CEP 21941-590
Rio de Janeiro - RJ
(+55)(21)22908091
thalitamoraes@gmail.com

RESUMO
As áreas de pesquisa em CSCW e Groupware vêm apresentando diversas propostas e soluções de apoio à colaboração. Atualmente, um dos grandes desafios está em atestar o quanto estas propostas se mostram efetivas em situações reais. Neste sentido, foi proposto o projeto CSCWLab, cujo objetivo principal é definir uma estrutura conceitual para o projeto de avaliações de sistemas colaborativos. A base desta estrutura é uma ontologia, que organiza os principais conceitos e elementos de avaliações de sistemas colaborativos, servindo como uma guia para pesquisadores no planejamento e execução de avaliações.

ABSTRACT
CSCW and Groupware research areas have been presenting many applications to support collaboration. Currently, one of the great challenges is in attesting how effective these proposals are in real situations. The CSCWLab Project presents a conceptual structure for collaborative systems evaluations planning. The base of this structure is an ontology that organizes the main concepts and elements, serving as a guide for researchers in planning and performance of such evaluations.

Palavras-chave
CSCW, groupware, avaliações, gestão de conhecimento, ontologia.

Keywords
CSCW, groupware, evaluation, knowledge management, ontology.

1 INTRODUÇÃO
A avaliação de sistemas colaborativos tem se apresentado como uma questão de pesquisa relevante. Nos últimos anos, as áreas de pesquisa em CSCW e groupware deram origem a diversas propostas e soluções de apoio à colaboração.

Atualmente, um dos grandes desafios da área está em atestar o quanto estas propostas se mostram efetivas em situações reais.

A necessidade de se estudar mais detalhadamente as questões de avaliação em groupware pode ser observada pelas publicações e conferências devotadas a este tema [3]. Pode-se dizer que ainda nos encontramos longe do consenso sobre como a avaliação de sistemas colaborativos pode ser conduzida, apesar de propostas terem sido apresentadas [1][2][3] [4][5][6].

Dentro do contexto de grupos de pesquisa nesta área, ferramentas, ambientes e protótipos são constantemente especificados e desenvolvidos como resultado de trabalhos de graduação, pós-graduação e pesquisa. A principal dificuldade enfrentada por estes grupos está em como realizar a avaliação destes produtos de forma a verificar a extensão de sua aplicação e apoio à atividade colaborativa que endereça. Qual o escopo da avaliação, como definir variáveis para medição de resultados, que instrumentos devem ser aplicados, como se seleciona participantes e se configura os grupos que tomarão parte da avaliação, até onde se deve ir para validar a hipótese proposta?

Neste sentido, foi proposto o projeto de pesquisa CSCWLab [7][8], cujo objetivo principal é definir uma estrutura conceitual para o projeto de avaliações de sistemas colaborativos. Como base desta estrutura conceitual, o CSCWLab propõe a definição de uma ontologia para avaliações de sistemas colaborativos que organiza os principais conceitos e elementos de avaliações deste tipo de sistemas, servindo como uma guia para pesquisadores no planejamento de avaliações [8][9].

Por outra perspectiva, percebe-se também a dificuldade em se acumular conhecimento sobre o uso de sistemas colaborativos de forma a estabelecer conclusões sobre sua aplicação em contextos específicos. O desempenho de um sistema colaborativo depende da variedade de personalidades e comportamento dos membros dos grupos que o utiliza, o efeito das dinâmicas sociais, motivacionais, econômicas e políticas. Todas estas questões interferem na maneira que pessoas se utilizam de sistemas colaborativos, tornando difícil ou quase impossível obter controle sobre todas as variáveis relacionadas. Avaliação de groupware, portanto, é reconhecidamente uma atividade custosa com geração de resultados com baixo potencial de generalização.

SBSC Simpósio Brasileiro de Sistemas Colaborativos
20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 78-86.

Sem os mecanismos de avaliação apropriados, a comunidade de desenvolvimento de sistemas colaborativos não é capaz de acumular experiência suficiente para aprender e construir sistemas com maior qualidade [10]. Neste sentido, como evolução e continuidade do projeto CSCWLab, sugere-se a construção de uma base de conhecimento, organizada utilizando-se a ontologia proposta, para a organização colaborativa de planos e resultados de avaliações realizadas por pesquisadores. Esta base de conhecimento acumularia, experiências, resultados e reflexões sobre o planejamento e condução de avaliações, permitindo que o conhecimento obtido durante a execução desta tarefa não se perca e possa ser utilizado por outros pesquisadores e desenvolvedores da área em seus projetos de aplicações ou planejamento de novas avaliações.

O objetivo deste artigo é apresentar a especificação e desenvolvimento desta base de conhecimento, bem como do sistema que permite seu gerenciamento. Para tal, este artigo organiza-se em 5 seções. Na Seção 2 apresentamos a proposta do projeto CSCWLab para orientação na avaliação de sistemas colaborativos. A Seção 3 descreve a ontologia definida para organizar o conhecimento gerado por pesquisadores no uso do CSCWLab para planejamento e execução de avaliações. Na seção 4 é apresentada a implementação e disponibilização da base de conhecimento, visando o registro e a recuperação de informações. A Seção 5 conclui o artigo e apresenta trabalhos para desenvolvimento futuro.

2 CSCWLab

A primeira preocupação do pojeto CSCWLab foi focada na definição de um método para avaliação de groupware que identificasse os conceitos e dimensões que deveriam ser consideradas durante a avaliação. Durante a avaliação de groupware, podemos nos direcionar para: avaliar e descrever o contexto sobre o qual a aplicação será utilizada; avaliar forças e fraquezas da usabilidade da aplicação; avaliar o nível de colaboração atingido durante o uso da aplicação; e avaliar o impacto cultural e tecnológico alcançado com ela com o tempo. Cada dimensão (Figura 1) deve ser considerada como passos subseqüentes de um método de condução de avaliações-piloto [7].

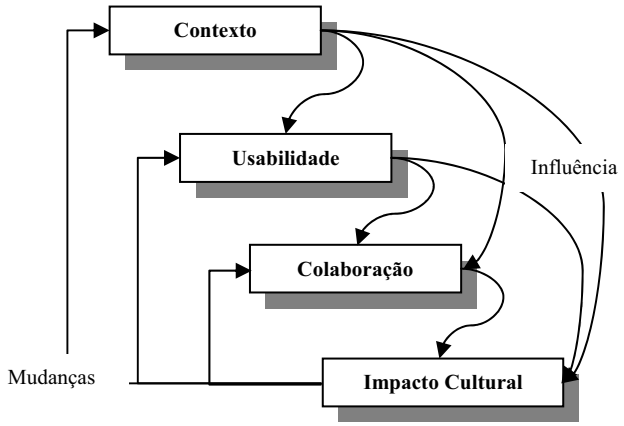


Figura 1 – Dimensões da avaliação de Groupware

Essas quatro dimensões têm um relacionamento estreito com cada um dos outros termos de avaliação e foram consideradas e analisadas para a construção da ontologia de domínio implementada no ambiente do CSCW Lab.

Contexto do grupo. Grupos são completamente originais, ou seja, é quase impossível encontrar dois grupos que apresentem os mesmos valores para atender a variáveis independentes em experimentos. Além disso, frequentemente não encontramos o grupo "ideal" para conduzir avaliações. Encontrar ou construir grupos para a avaliação é difícil e caro. Uma caracterização do grupo observado é essencial para a interpretação e a discussão do resultado da avaliação. O contexto do grupo influencia todas as outras dimensões. Por exemplo, se o grupo for altamente comprometido para executar uma tarefa, é possível que supere problemas de usabilidade. O nível de colaboração será elevado, desde que os membros estejam dispostos a trabalharem juntos e, finalmente, é também possível que a ferramenta tenha um impacto positivo no ambiente do trabalho ou da interação.

Usabilidade. A usabilidade de uma aplicação determina seu uso apropriado e também sua aceitação por usuários. A avaliação de usabilidade de groupware é uma tarefa complexa uma vez que envolve não somente a avaliação de como um usuário pode usar a ferramenta, mas também se o grupo pode interagir como esperado usando a ferramenta.

Nível de colaboração. A colaboração pode ocorrer em muitos níveis e depende da natureza e dos objetivos da tarefa do grupo. Para avaliar a colaboração é necessário especificar as medidas ou variáveis que determinam como um grupo colabora. Por exemplo, em um fórum de discussão, uma possibilidade para medir a colaboração é contar o número das contribuições geradas pelo grupo. Entretanto, a colaboração em um fórum é somente eficaz se as contribuições forem não somente introduzidas, mas lidas por outros participantes. Através de instrumentos como questionários ou pela observação direta, os avaliadores podem também perceber a satisfação e a sensação dos participantes sobre a colaboração que ocorre entre eles.

Impacto Cultural. Uma tecnologia pode ter o uso completamente diferente em contextos diferentes. A dimensão final da avaliação do grupo é como uma ferramenta do groupware muda a maneira como as pessoas ou o grupo trabalham, as formas esperadas e as inesperadas de uso da ferramenta, e também como a organização ou o grupo começa a sentir ou reconhecer o que a ferramenta nova introduziu em sua cultura.

A segunda questão tratada pelo CSCW Lab é o estabelecimento de um ambiente para armazenamento e recuperação de dados sobre avaliações realizadas em diversos contextos. A inclusão destas informações em uma base deve ser feita segundo um modelo, que define os elementos desde o projeto da experimentação conduzida até os resultados e análise realizados.

O objetivo deste ambiente é tornar-se um repositório compartilhado de conhecimento sobre avaliações em groupware. Pesquisadores da área podem acessar esta base, para obter suporte na a construção do seu plano de experimentação; e observar resultados de outras experiências, comparando com os seus próprios.

3 A ONTOLOGIA

A estratégia escolhida para construir a estruturação do repositório ou base de conhecimento das avaliações de groupware foi definida e populada em uma ontologia de domínio [17]. Através da ontologia desenvolvida, criamos um modelo normativo do CSCW Lab e uma semântica para o mesmo. As dimensões do CSCW Lab são o núcleo dessa ontologia, associada com outros conceitos comuns que existem nas avaliações em geral:

instrumentos, dados, produtos, interpretações de resultados e outros.

O desenvolvimento de uma ontologia é um processo iterativo, que inclui ciclos de: definição de classes, organização das classes numa hierarquia taxonômica, definição de atributos (slots) e descrição de valores permitidos (instâncias) para eles.

Para elaboração da ontologia da base de conhecimento do CSCW Lab, realizamos um brainstorming, seguido de uma seleção mais criteriosa e conseqüente refino de uma lista dos termos que gostaríamos de abordar ou explicar a um usuário do CSCW Lab. A partir dos termos presentes na lista gerada realizamos algumas reuniões para a criação das classes que representariam esses termos, apoiados no primeiro esboço da ontologia de domínio para avaliação de groupware proposto em [9]. A modelagem foi feita com apoio da ferramenta Protege.

As classes foram organizadas dentro de uma taxonomia hierárquica, utilizando-sw uma abordagem *top-down*: em um primeiro momento, consideramos os conceitos mais gerais do domínio e depois fizemos a especialização subseqüente dos mesmos.

Uma visão geral da ontologia do CSCWLab pode ser obervada na Figura 2. A classe Avaliação é a principal classe da ontologia CSCW Lab. Ela se refere à forma como a avaliação da ferramenta de groupware vai ser conduzida, relacionando-se, através de seus atributos, com as seguintes classes:

- Ferramenta de Groupware: objeto de avaliação;
- Plano de Avaliação, que é o detalhamento de como será conduzida;
- Resultados, que concentra os dados brutos obtidos após a execução do plano de avaliação e da validação da hipótese;
- Questões, que contém os desdobramentos decorrentes da afirmação da hipótese;
- Dimensão, que apresenta as formas nas quais a avaliação pode ser direcionada ao ser realizada (Contexto do Grupo, Usabilidade, Nível de Colaboração, Impacto Cultural).

A classe Plano de Avaliação possui atributos que explicam como a avaliação será realizada.. Seus relacionamentos são com as seguintes classes:

- Cronograma, onde são organizadas as tarefas e prazos para a realização da avaliação;
- Cenário de Avaliação, que descreve o conjunto de características do ambiente onde a ferramenta de groupware está sendo utilizada e avaliada;
- Grupo, que representa o conjunto de pessoas que participaram dos experimentos propostos pelo plano de avaliação da ferramenta;

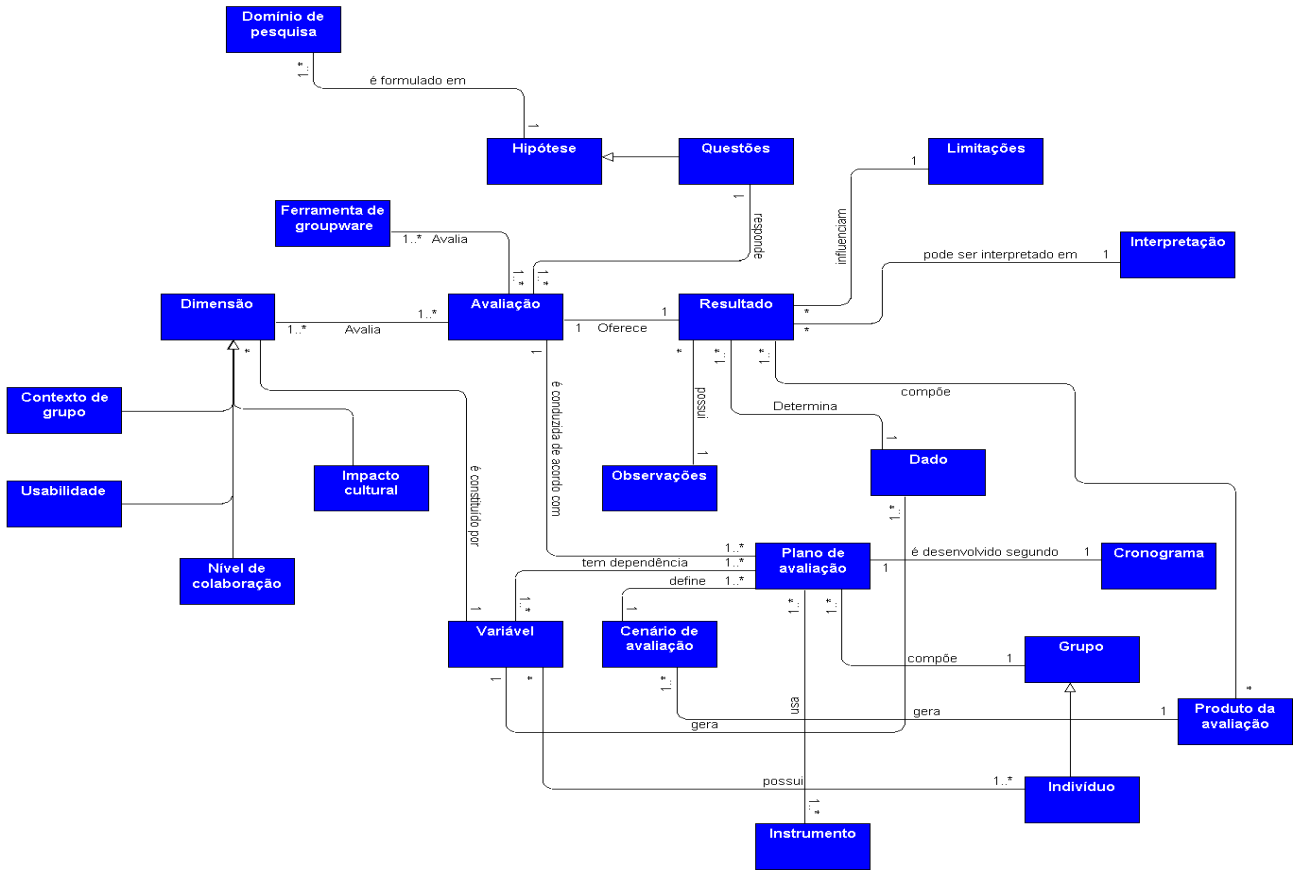


Figura 2 – Ontologia CSCW Lab

- Instrumento, que representa os instrumentos utilizados para avaliar o objeto da avaliação;
- Variável, que representa as variáveis utilizadas na medição qualitativa ou quantitativa das dimensões da avaliação.

As demais classes do modelo são:

- Domínio de Pesquisa, que representa o campo de atuação onde o pesquisador situa a sua pesquisa e/ou experimento.
- Hipótese, que representa uma conjectura específica a ser validada dentro de um domínio de pesquisa.
- Dado, que representa as informações obtidas durante o experimento e que irão auxiliar na definição de um resultado para validação de uma hipótese.
- Interpretação, que representa a explicação inferida sobre os resultados obtidos após a realização da avaliação da ferramenta.
- Limitações, que representa os problemas que possam aparecer durante o processo.
- Observações, que representa notas feitas pelo avaliador durante a realização do projeto de avaliação.
- Produto, que representa informações obtidas através dos resultados finais de uma avaliação.

As classes especificadas neste modelo representam as entidades estabelecidas na metodologia da pesquisa científica, aplicadas ao contexto genérico de uso de groupware. Enquanto um modelo de conhecimento não é completo, pelo contrário entende-se que deve estar em constante aprimoramento.

Na medida em que cada tipo de avaliação deverá acontecer em um domínio específico, e portanto com requisitos de planejamento e instrumentos próprios, esta ontologia deverá evoluir, tanto no sentido de incorporar novas entidades, quanto relacionamento. A intenção é que o projeto se desenvolva no sentido de tornar-se um fórum aberto para critica e agregação por parte da comunidade de pesquisa.

4 GESTÃO DE CONHECIMENTO NO CSCWLab

O objetivo da criação da base de conhecimento é gerar um ambiente que poderá fornecer aos pesquisadores, informações sobre as avaliações de groupware já realizadas. Sendo assim, será possível a troca de informações e experiências no assunto, visando a criação de um consenso, ou padrão, a respeito da metodologia a ser aplicada para sua realização da avaliação. Para a construção desta base de conhecimento, optou-se pela implementação da ontologia em um ambiente que permite a captura, representação e recuperação de informações sobre o planejamento e condução de avaliações [12].

Entende-se que a captura, representação e recuperação são as fases principais do processo de gestão de conhecimento [18]. A ontologia definida estrutura a base de conhecimento do CSCWLab bem como define um guia para o registro/captura de conhecimento sobre avaliações e sua posterior recuperação.

4.1 Implementação da Ontologia

Para o desenvolvimento da base de conhecimento, foram efetuadas avaliações das ferramentas de construção, publicação e gestão de conteúdo baseado em ontologias. Como resultado desta análise, optou-se pela ferramenta Protegé [11], desenvolvida pela Universidade de Stanford, por ser a mais consolidada, bem documentada e que oferecia a possibilidade de publicação de conteúdo na Internet.

No Protegé, os conceitos da ontologia foram definidos como classes e suas estruturas internas, bem como relacionamentos foram definidos através de *slots.* O *slot* pode representar um valor simples, como números inteiros e reais, cadeias de caracteres, valores lógicos ou uma referência a uma instância; este último tipo foi usado para definição de relacionamentos entre classes e instâncias. Os atributos da ontologia do CSCW Lab foram classificados em extrínsecos, intrínsecos ou ainda de relacionamentos com outras classes ou atributos. Em seguida, foram definidos os tipos de valor dos atributos, os relacionamentos entre as classes e suas cardinalidades.

Para disponibilizar a base de conhecimento através da web, foi realizada a configuração do ambiente de implementação do *plug-in* Protégé Web Browser. Assim, foi configurado o ambiente para publicação da ontologia na web [13] (Figura 3)

4.2 Capturando conhecimento sobre avaliações

Para a captura de informações sobre avaliações espera-se que os pesquisadores da área, ao realizarem suas avaliações, possam registrar na base CSCWLab o planejamento das avaliações realizadas, bem como os resultados atingidos. O pesquisador poderá navegar pela ontologia disponibilizada, instanciando suas classes, seus atributos e relacionamentos. Alguns contextos de registro de informação sobre avaliações podem ser sugeridos:

- Avaliações de protótipos de projetos de pesquisa: em um projeto de pesquisa, avaliações planejadas de diferentes versões de um protótipo ou ferramenta em desenvolvimento podem ser registradas no CSCWLab. Este registro pode ser útil não só para a equipe de pesquisadores do projeto como para a comunidade em geral que passa a ter acesso aos resultados obtidos.
- Avaliações de protótipos de trabalhos de mestrado/doutorado: dissertações de mestrado e teses de doutorado na área de CSCW comumente incluem a especificação e desenvolvimento de protótipos de ferramentas/ambientes cuja validação passa pela necessidade de realização de avaliações. O registro do plano e resultados destas avaliações na base CSCWLab, além do registro usual nos textos das dissertações e teses, ajudam a estruturar o conhecimento do grupo de pesquisa onde os trabalhos estão inseridos.
- Avaliações comerciais: organizações ou consultores em sistemas colaborativos podem registrar na base CSCWLab informações sobre avaliações contínuas sobre o uso de uma determinada ferramenta em um dado contexto organizacional, compondo um histórico e conhecimento sobre o uso interno da ferramenta, ajudando na compreensão e análise de seu uso. Ressalta-se aqui, para fins de privacidade de informação, que

organizações podem manter bases CSCWLab restritas ao uso interno de suas corporações.

4.2.1 Exemplo

Nesta seção, cabe a exemplificação do registro de informações na base CSCWLab, com o intuito de ilustrar os componentes da ontologia definida. Após a criação da ontologia CSCW Lab no Protegé e de sua disponibilização na web, foram incluídos dados de três avaliações realizadas no contexto de dissertações de mestrado em sistemas colaborativos [14][15][16], gerando as primeiras instâncias na base de conhecimento. Este processo foi realizado com o objetivo de validar a ontologia criada bem como o ambiente construído. Para ilustrar os passos necessários para se publicar as avaliações na base de conhecimento CSCWLab, tomamos como exemplo a avaliação realizada em um dos trabalhos de pesquisa mencionados - "*A Reengenharia Participativa apoiada por uma ferramenta de groupware: CEPE, um editor cooperativo para a elicitação de processos*" [16].

Ressalta-se que apesar da avaliação ter sido realizada anteriormente à disponibilização do ambiente, as informações solicitadas pela base de conhecimento, com base na ontologia definida, auxiliam aos pesquisadores na definição e planejamento de suas avaliações. Simularemos, portanto, nesta ilustração, o fato de que a inserção de informações na base de conhecimento tivesse sido feita pelo próprio pesquisador que realizou a avaliação.

Na ontologia do CSCW Lab, a classe que pode ser chamada de inicial, é a classe "*Domínio de Pesquisa*". Portanto, esta classe é a primeira a ser definida pelo pesquisador. No caso da pesquisa deste exemplo, o domínio de pesquisa é a "*Reengenharia Processos*" e a ferramenta analisada é o "*CEPE*". O CEPE é descrito como: "*CEPE (versão 1). Implementada usando o groupkit como ambiente de desenvolvimento e a linguagem Tcl TK*", e também o seu objetivo: "*Sistema para edição cooperativa de textos*", juntamente com o público alvo: "*Qualquer tipo de empresa disposta em efetuar uma reengenharia de processos*" e o tipo da ferramenta: "*Comunicação*".

Outro ponto importante dentro da ontologia CSCW Lab é o registro da hipótese sendo observada pela pesquisa em questão - "*O enfoque participativo (como método de trabalho) juntamente com uma ferramenta computacional para apoiar a elicitação de processos pode melhorar a tarefa de elicitar processos*". Uma hipótese pertence a um domínio de pesquisa. Por isso, esta classe possui um relacionamento com a classe "*Domínio de pesquisa*", relacionamento este representado pelo campo "*contida_em*". Neste caso, foi relacionada a hipótese acima com o domínio de pesquisa "*Reengenharia Processos*".

Toda hipótese gera uma série de questões que a subdividem ou uma questão que a explica de forma mais clara. Neste exemplo, a hipótese é detalhada na seguinte questão: "*Como fazer a elicitação cooperativa do conhecimento que se encontra distribuído, tratando as diferentes visões que se tem do mesmo processo, buscando uma documentação para o mesmo ?*".

Neste ponto, torna-se necessária a verificação de que classe deverá ser preenchida primeiro. Logo, é importante diferenciar as classes, para que se facilite a população da base. Para isto, foram criadas duas nomenclaturas que serão utilizadas, as "*classes internas*" e as "*classes externas*".

As "*classes internas*" são aquelas que além de possuírem as informações principais e centrais dentro da ontologia, são as que possuem os campos (atributos) que fazem os relacionamentos com as outras classes. As "*classes externas"* são classes mais específicas, que geralmente se relacionam com apenas uma "*classe interna",* enquanto estas se relacionam com mais de uma classe "*externa*" ou "*interna*". Por exemplo, classes como "*Interpretação*" e "*Ferramenta de Groupware*" são "*classes externas",* já classes como "*Plano de Avaliação*" e "*Resultado*" são chamadas de "*classes internas".*

Portanto, a melhor decisão neste ponto é instanciar as "*classes externas".* Desta forma, ao chegarmos no ponto da instanciação das "*classes internas",* já teremos dados suficientes para efetuar os relacionamentos sem problemas.

Sendo assim, o pesquisador é questionado quanto a buscar informações sobre uma destas "*classes externas",* a classe "*Limitações*". Apesar de ser um dos últimos itens a serem tratados não apenas neste, mas na maioria das dissertações analisadas, iremos verificá-la pelo motivo citado anteriormente. No contexto da pequisad tomada como exemplo, o pesquisador informaria que foram constatadas as seguintes limitações: pouco tempo disponível para a tarefa proposta; houve somente um estudo de caso; impossibilidade de tratar todas as variáveis envolvidas.

A partir deste momento, existem informações suficientes para criar o plano de avaliação, que tem por objetivo melhorar a organização e definição da avaliação a ser conduzida. Para criar o plano de avaliação, segundo a ontologia CSCW Lab, idealmente, é necessário saber qual grupo estará participando da avaliação, quais variáveis foram criadas para a avaliação, em que cenário a avaliação foi realizada, qual o cronograma definido para a avaliação e qual o instrumento utilizado por ela. Desta forma, foi utilizado o conteúdo instanciado nas classes "*Grupo*", "*Variável*", "*Cenário da Avaliação*", "*Cronograma*" e "*Instrumento*" para a criação dos relacionamentos entre estas classes e a classe "*Plano de Avaliação*".

Em seguida, verifica-se a inclusão de informações relacionadas ao plano da avaliação. A classe "*Instrumento*", neste exemplo, será preenchida com a informação de que a metodologia para a avaliação foi feita através de questionários. Quanto à instancia a classe "*Cronograma*", o pesquisador informa que o cronograma foi realizado em duas fases, com o período de duração de quatro dias cada. A primeira fase foi realizada sem a ferramenta e o enfoque foi na participação, e foram estabelecidas rotinas de entrevistas individuais dos membros da equipe pelo consultor. Já a segunda fase foi realizada com a ferramenta e o enfoque também foi participativo.

A classe "*Variável*" representa as variáveis criadas e utilizadas na medição qualitativa ou quantitativa das dimensões de uma avaliação: quantidade de mensagens trocadas; quantidade de contribuições na construção de um produto coletivo; quantidade/inferência sobre as contribuições de outros membros do grupo; presença de liderança; envolvimento com a definição do processo; cumprimento das tarefas; entendimento das tarefas e suas correlações.

A classe "*Dimensão*" é uma superclasse das subclasses "*Contexto de grupo*", "*Impacto cultural*", "*Nível de colaboração*" e "*Usabilidade*", que são relacionadas diretamente às variáveis criadas para a avaliação da ferramenta de groupware, já que estas são utilizadas na avaliação das dimensões. Neste momento, o pesquisador pode definir quais dimensões serão endereçadas em sua avaliação e associar variáveis para sua medição.

Criado o plano de avaliação, chega ao momento de verificar a avaliação propriamente dita. A avaliação é conduzida utilizando-se um plano de avaliação como base na sua execução. Além do plano, é relacionada, a ferramenta que será avaliada, as dimensões que são utilizadas dentro da avaliação e as questões que são levantadas para serem respondidas por ela. Cada avaliação possui um objetivo próprio e um tipo, que pode ser, por exemplo, um experimento, como é o caso da dissertação citada neste estudo de caso. Ao final da avaliação, é possível registrar os seus resultados, que são originados através da execução da avaliação e podem ser influenciados pelas limitações existentes; eles estão ainda, associados ao produto da avaliação e aos dados criados por ela.

Cada resultado obtido pode ter a sua própria interpretação e observações. Desta forma, foi utilizado o conteúdo das instâncias das classes "*Avaliação*", "*Limitações*", "*Produto da Avaliação*", "*Dados*", "*Interpretação*" e "*Observações*" para a criação do relacionamento entre os dados destas classes com a classe "Resultado". O pesquisador poderia, por exemplo, registrar as seguintes conclusões: "*A equipe relatou uma sensação de melhoria nas interações e no entendimento sobre as atividades executadas dentro da empresa*" e "*A equipe utilizou pouco o chat*". A partir da análise destes resultados, as seguintes interpretações dos resultados podem ser registradas: "*Houve um enriquecimento do relacionamento entre as contribuições individuais*" e "*Talvez por tratar-se de pessoas não ligadas à tecnologia e por desconhecerem os mecanismos que poderiam ser empregados*".

Ao final da inserção de instâncias na ontologia CSCW Lab, foi verificado que nem todas as classes tiveram seus atributos "preenchidos". Foi verificado que cada avaliador efetua o seu trabalho seguindo seus próprios instintos e experiências. Desta forma, é dificultada a pesquisa e a definição de qual a melhor forma de se fazer uma avaliação. Podemos verificar então a importância da é fornecer uma ontologia que possa apoiar pesquisadores e desenvolvedores de groupware na modelagem e construção de suas ferramentas.

Para ilustrarmos a interface com a ferramenta para inclusão de dados na base de conhecimento, a Figura 4 apresenta o exemplo do registro de uma instância da classe "*Avaliação*" onde o campo "*Objetivo Da Avaliação*" é um atributo texto, de preenchimento normal, mas o atributo "*Conduzida De Acordo Com*" (de relacionamento com a classe "*Plano de Avaliação*") e o atributo "*Faz Uso De*" (de relacionamento com a superclasse "*Dimensão*") são atributos de relacionamento com outras classes.

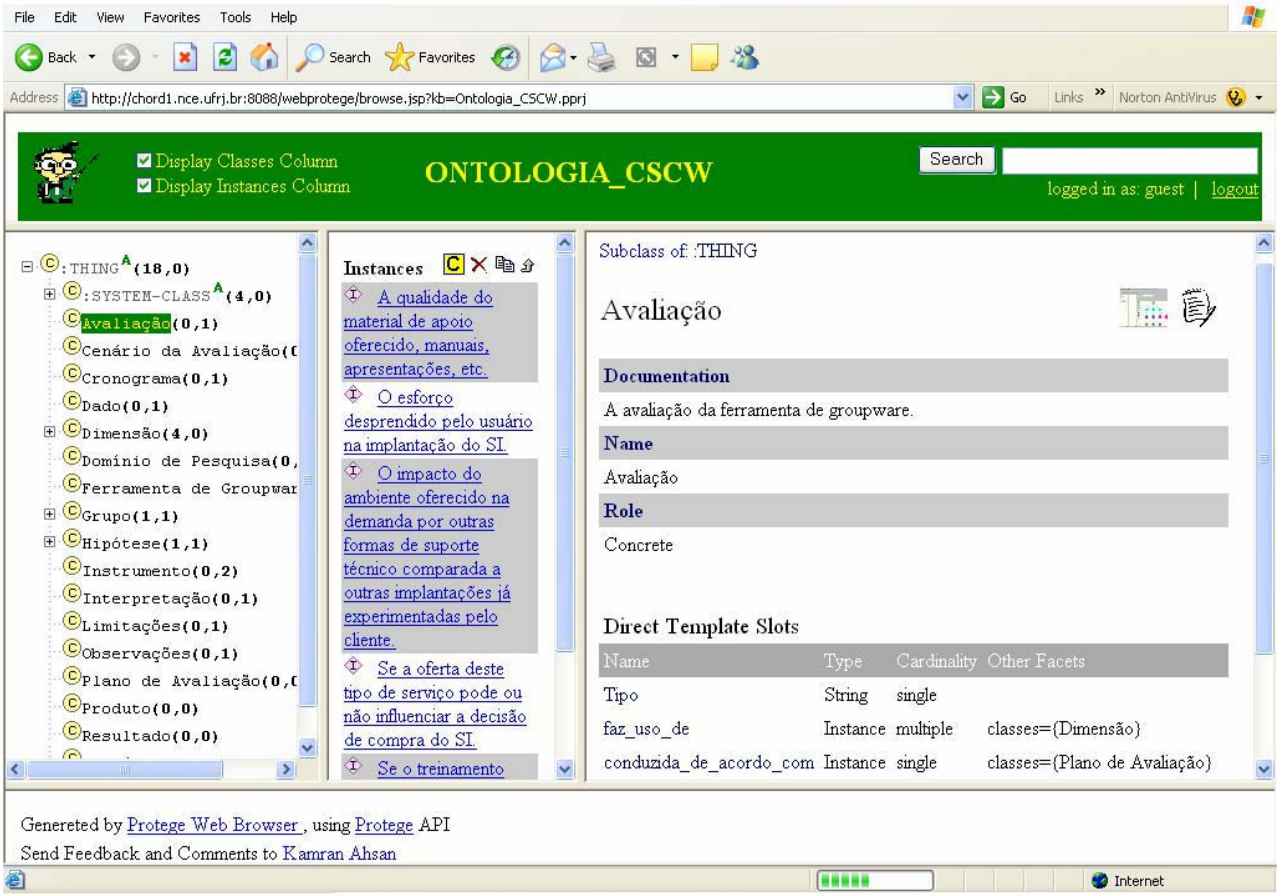


Figura 3 – Interface Web da Ontologia CSCW Lab

4.3 Recuperando conhecimento sobre avaliações

Para recuperar informações, o pesquisador poderá navegar pelas classes, seus atributos e relacionamentos, consultando informações sobre as avaliações registradas e, consequentemente, o planejamento e resultados de avaliações realizadas por outros pesquisadores. Alguns cenários de consultas e navegação pela base podem ser enumerados:

- Planejamento de avaliações: para planejar a avaliação de seu trabalho de pesquisa, o pesquisador pode consultar a base em busca de avaliações realizadas dentro do mesmo domínio; avaliações neste ou em outro domínio utilizando a mesma ferramenta; avaliações que exploraram a mesma dimensão de avaliação; variáveis definidas etc.

- Comparação de resultados: após concluir sua avaliação, registrando-a na base de conhecimento do CSCWLab, o pesquisador pode, a fim de complementar seu trabalho de pesquisa e argumentação da validação de sua hipótese, fazer uso das observações, resultados e interpretações de outras avaliações para comparar com os resultados obtidos, enriquecendo sua própria análise.

- Especificação de sistemas: pesquisadores com vistas a especificar novos sistemas de apoio à colaboração podem navegar pela base de conhecimento do CSCWLab, analisando os resultados obtidos com as avaliações em um determinado domínio, subdomínio ou questão, servindo as conclusões aí registradas como um corpo de conhecimento inicial sobre o que funciona e o que não funciona, ajudando-o a determinar requisitos específicos para o sistema em desenvolvimento.

Percebe-se, no entanto, que a implementação atual da base de conhecimento do CSCWLab permite exclusivamente a navegação pelas informações contidas na base. Ela poderá ser ainda mais útil aos pesquisadores quando oferecer a possibilidade de realizar consultas baseadas na semântica definida pela ontologia para inferir conclusões sobre o conhecimento do grupo ou comunidade de pesquisa em questão. A possibilidade de inferência e a qualidade e efetividade dos resultados dependerá, ainda, que a base de conhecimento armazene um volume de avaliações suficiente para sua realização.

Esse seria um grande ganho para os pesquisadores pois pode-se diminuir a dificuldade de avaliar dentro da área, assim como fornecer uma metodologia que facilite comparações dos resultados obtidos pelos mesmos em seus estudos, garantindo-os como relevantes para a interpretação.

Há dificuldade em se obter informações mais específicas sobre os efeitos e conquistas do groupware, de caminhos que já foram seguidos e quais se adequam mais às necessidades e realidade de pesquisa deste ou daquele pesquisador. Em um dos trabalhos utilizados como fonte para extração de dados para popular a ontologia, o autor relata que não atingiu o objetivo pretendido de forma satisfatória através da avaliação que realizou. Imaginamos que, para este caso, relatos de experimentações anteriores poderiam ter ajudado no desenho do seu plano de avaliação, auxiliando na obtenção de variáveis mais precisas e úteis para validação da hipótese defendida pelo mesmo.

5 CONCLUSÃO

A proposta do CSCW Lab é fornecer um ambiente baseado em ontologia que possa apoiar pesquisadores e desenvolvedores de sistemas colaborativos no processo de testes e avaliações de suas ferramentas, permitindo que o conhecimento obtido durante a execução desta tarefa não se perca e possa ser utilizado por outros pesquisadores e desenvolvedores da área em melhorias e trabalhos futuros.

Além disso, o CSCW Lab visa fornecer uma metodologia de avaliação mais formal e homogênea que facilite o processo de planejamento, construção e testes de utilização de sistemas colaborativos, tornando-os melhores, sobretudo em qualidade e dessa forma aumentando a utilização destes sistemas pelas organizações na execução de suas tarefas.

Espera-se com esta proposta ampliar a discussão sobre o uso de groupware e obter resultados mais sólidos de pesquisa, através do compartilhamento dos problemas e soluções observados nas experimentações.

O exemplo construído mostrou o caminho de uso do ambiente no contexto da gestão de conhecimento em CSCW através do repositório. Percebe-se a sinalização de questões que podem ter ficado em aberto ao longo da pesquisa com uma ferramenta, ou seja, ao criar instâncias a partir de experimentos já realizados, o pesquisador é levado a refletir sobre seus resultados.

As buscas podem ser feitas sobre planejamentos similares ou resultados que possam ser complementares. Então, a dificuldade de replicação de experimentos é parcialmente minimizada, pois o pesquisador pode partir dos indícios descobertos em outras pesquisas.

Como trabalhos futuros, podemos enumerar: a evolução da interface do ambiente de gestão de conhecimento para facilitar a inclusão de informações de avaliações e direcionar ainda mais o planejamento de avaliações; a discussão e proposta de heurísticas para inferências com base no conhecimento coletado; a divulgação e motivação aos pesquisadores da área para contribuir para seu corpo de conhecimento; e a análise qualitativa do uso da base de conhecimento pelos pesquisadores.

6 REFERÊNCIAS

[1] Pinelle, D. 2000 *A Survey of Groupware Evaluations in CSCW Proceedings*. Research Report 00-1, Computer Science Department, University of Saskatchewan

[2] Pinelle, D., and Gutwin, C., 2000, A Review of Groupware Evaluations. In *Proceedings of Ninth IEEE WETICE 2000 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Gaithersburg, Maryland, Junho.

[3] Nutter, D., Boldyreff, C., 2005, *Workshop on Evaluating Collaborative Information Systems and Support for Virtual Enterprises – Workshop Report*, as part of the 9th Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Junho, National Institute of Standards and Technology, USA. (relatório disponível em: hemswell.lincoln.ac.uk/%7Ednutter/wetice05-ece/index.php)

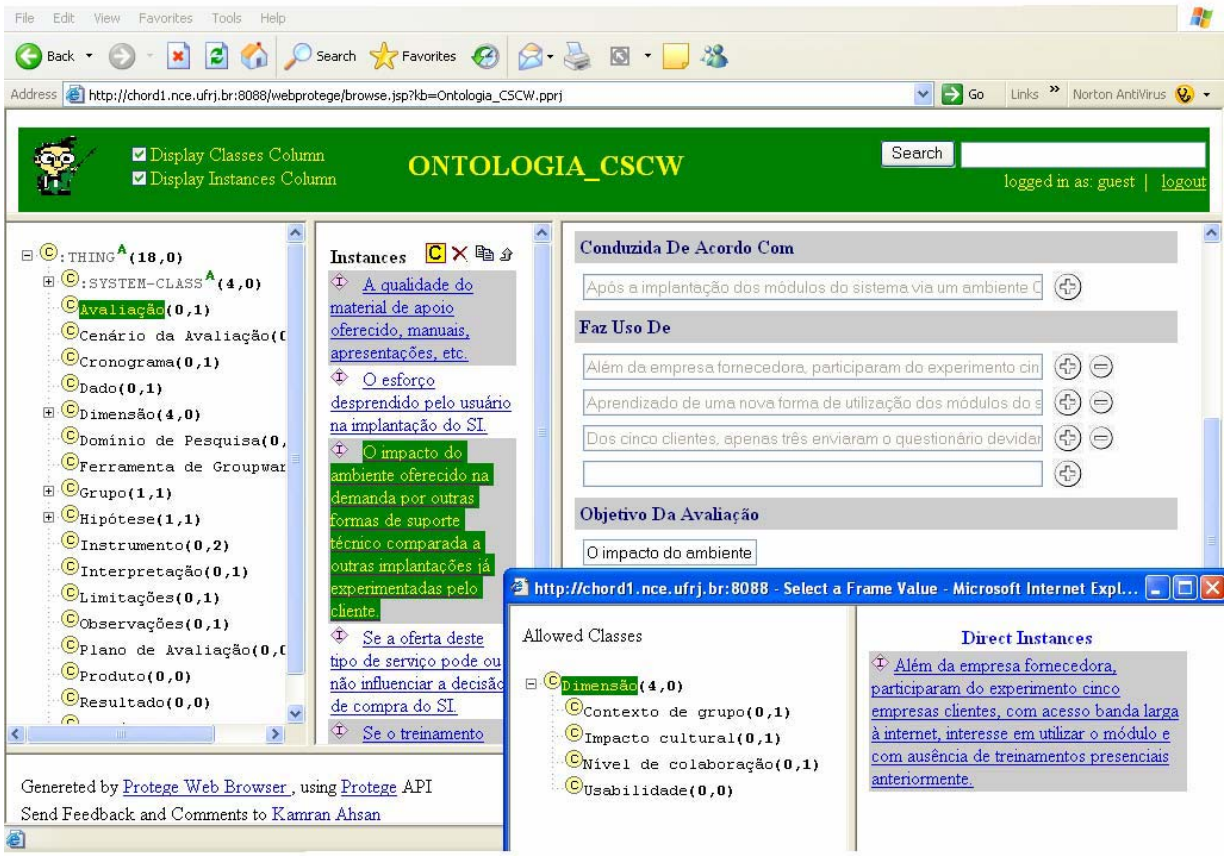


Figura 4 - Tela de criação de registros

[4] McGrath, J.E., Methodology Matters: Doing Research in the Behavioral and Social Sciences. In *Readings in Human-Computer Interaction: Toward the Year 2000*, 2nd Ed. Baeker, R.M., Grudin, J., Buxton, W.A.S., Greenberg, S. (eds.), Morgan Kaufman Publishers, San Francisco, 1995, 152-169.

[5] Ramage, M., Twidale, M., Sommerville, I., *Evaluation of Cooperative Systems Project*, (http://www.comp.lancs.ac.uk/computing/research/cseg/projects/evaluation/)

[6] Dennis, A., Valacich, J.S., 2001, Conducting Research in Information Systems, *Communications of the Association for Information Systems*, Vol. 7, Article 5, July.

[7] Araujo, R.M., Santoro, F.M., Borges, M.R.S. 2002, The CSCWLab for Groupware Evaluation. In *Proceedings of the 8th International Workshop on Groupware*, Lecture Notes in Computer Science, La Serena, Chile, v. 2440, p. 222-231 2002.

[8] Araujo, R. M., Santoro, F. M., Borges, M. R. S., A conceptual framework for designing and conducting groupware evaluations. In *Int J Of Computer Applications In Technology*, Inglaterra, p. 139-150, 2004.

[9] Araujo, R. M.; Santoro, F. M. ; Borges, M. R. S., The CSCW Lab Ontology for Groupware Evaluation. In: *Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design*, 2004. v. II. p. 611-615.

[10] Baker, K., Greenberg, S., Gutwin, S., 2001, Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. In: *Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'01)*, Toronto, Canadá.

[11] Protegé. http://protege.stanford.edu/

[12] Moraes, T.M.S., Prata, F.R., 2006, CSCW Lab – Uma ontologia para avaliação de groupware, Projeto Final de Curso de Graduação, Departamento de Ciência da Computação, IM/UFRJ, Rio de Janeiro, RJ, Brasil.

[13]Base de Conhecimento CSCWLab - http://chord1.nce.ufrj.br:8088/webprotege

[14] Meire, A.P.; 2003, *Suporte à edição cooperativa de diagramas utilizando versões*. Dissertação (Mestrado em Informática) - UFRJ – Programa de Pós-Graduação em Informática - IM/NCE.

[15] Bittes, J.M.; 2004, *Um Método de Implantação de Sistemas Apoiado por Aprendizado Colaborativo*. Dissertação (Mestrado em Informática) - UFRJ – Programa de Pós-Graduação em Informática - IM/NCE.

[16] Freitas, R. M.; 2003, *A Reengenharia Participativa apoiada por uma ferramenta de groupware: CEPE, um editor cooperativo para elicitação de processos*, Dissertação de Mestrado, UFRJ – Programa de Pós-Graduação em Informática - IM/NCE.

[17] Guarino, N., 1997, Understanding, Building and Using

Ontologies. In: *Int. Journal Human-Computer Studies*, 46(2/3), February/March 1997.

[18] Abecker A., Aitken S., Schmalhofer F., Taitschian B., "KARATEKIT: Tools for the knowledge-creating company", in: Proceedings of 11th Workshop on Knowledge Acquisition, Modelling and Management KAW'98, Banff, 1998.

PatternFlow: Supporting Standardized Description and Enactment of Business Processes

Marcos Kalinowski
COPPE – Federal University of Rio de Janeiro
PO Box 68511 - 21 941-972
Rio de Janeiro - Brasil
mkali@cos.ufrj.br

Marcos R. S. Borges
NCE – Federal University of Rio de Janeiro
PO Box 2324 - 20 001-970
Rio de Janeiro - Brasil
mborges@nce.ufrj.br

Guilherme H. Travassos
COPPE – Federal University of Rio de Janeiro
PO Box 68511 - 21 941-972
Rio de Janeiro - Brasil
ght@cos.ufrj.br

RESUMO
O estabelecimento de padrões para tratar requisitos de workflow de forma sistemática é importante, uma vez que processos, após sua modelagem, deveriam poder ser descritos e executados por sistemas de gerenciamento de workflow. Apesar do surgimento de tais padrões, muitos deles não são apoiados por grande parte das linguagens de modelagem de workflow e dos sistemas de gerenciamento de workflow. Além disto, muitos de tais padrões apresentam soluções para problemas freqüentes durante a atividade de análise de projetos de workflow, o que realça a importância de apoiá-los.

Neste artigo apresentamos PatternFlow, uma infra-estrutura para descrever e executar processos de negócio cujo conjunto de requisitos foi derivado principalmente de conhecimento adquirido da análise de padrões de workflow. PatternFlow têm sido utilizada, entre outros, para implementar ISPIS, uma infra-estrutura para apoio ao processo de inspeção de software. Avaliações de ISPIS na academia e na indústria nos permitiram obter uma percepção adicional sobre a viabilidade de utilizar PatternFlow para executar processos em ambientes industriais.

ABSTRACT
The establishment of patterns to systematically address workflow requirements is important, since it should be possible for a modeled process to be described and enacted by workflow management systems. Despite the sprouting of those workflow patterns, many of them are not supported in a great amount of workflow modeling languages and workflow management systems. Moreover, many of those patterns recur quite frequently during the analysis phase of workflow projects, which highlights the importance of supporting such patterns.

In this paper we present PatternFlow, a framework for describing and enacting business processes whose requirements set was derived mainly from knowledge acquired from the analysis of documented workflow patterns. PatternFlow has been used, among others, to implement ISPIS, a framework for supporting the software inspection process. Evaluations of ISPIS in academy and industry allowed us to gain additional insight regarding the feasibility of using PatternFlow for enacting processes used in real industrial environments.

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 87-97.

PALAVRAS-CHAVE
Sistemas de Gerenciamento de Workflow, Padrões de Workflow, Descrição de Processos, Execução de Processos.

KEYWORDS
Workflow Management Systems, Workflow Patterns, Process Description, Process Enactment.

1. INTRODUCTION
The available workflow management systems present a large variety of languages, as well as concepts based on different paradigms [1]. Thus, understanding their main requirements and how well those requirements are supported by existing workflow management systems becomes a difficult task. One step towards facilitating this task was the initiative of establishing workflow patterns [1, 2, 3, and 4] to systematically address workflow management system requirements.

Since the sprouting of the workflow patterns [4], several workflow management systems have been evaluated considering them as reference. The evaluation of 15 commercial workflow management systems, for instance, is described in [1]. In this evaluation the authors noticed that each of the workflow patterns was supported in at least one of the systems. However, none of the systems offered support to all of the patterns. Moreover, most of them support only a relatively small subset of the more advanced patterns. Since many of those patterns recur quite frequently during the analysis phase of workflow projects [1], some business processes, depending on their complexity, can not be described nor enacted by any of those workflow management systems.

Based on this scenario, we present PatternFlow, a framework for describing and enacting business processes, whose requirements were derived mainly from the analysis of the documented workflow patterns. The initial concept of PatternFlow is described in [5]. Over the years PatternFlow evolved and has been used, among others, to implement ISPIS [6], a framework for supporting the software inspection process. Having a real industry process described in PatternFlow allowed us to perform evaluations, in academy and industry [7]. The feedback of those evaluations helped to gain additional insight into the suitability of PatternFlow for enacting processes.

The remainder of this paper is organized as follows. Section 2 provides an overview of the workflow patterns. Section 3 describes PatternFlow and how the workflow patterns were supported in it. Section 4 presents the case study of using PatternFlow for implementing ISPIS, as well as the evaluations

Table 1. Workflow Patterns. Adapted from [1].

Basic Control Flow Patterns
Sequence: An activity in a process is enabled after the completion of another activity in the same process.
Parallel Split: A point in the process where a single thread of control splits into multiple threads of control which can be executed in parallel.
Synchronization: A point in the process where multiple parallel subprocesses converge into one single thread of control.
Exclusive Choice: A point in the process where, based on a decision or workflow control data, one of several branches is chosen.
Simple Merge: A point in the process where two or more alternative branches come together without synchronization. It is an assumption of this pattern that none of the alternative branches is ever executed in parallel.
Advanced Branching and Synchronization Patterns
Multiple Choice: A point in the process where, based on a decision or workflow control data, a number of branches are chosen.
Synchronizing Merge: A point in the process where multiple paths converge into one single thread. If more than one path is taken, synchronization of the active threads needs to take place.
Multiple Merge: A point in a process where two or more branches reconverge without synchronization. If more than one branch gets activated, possibly concurrently, the activity following the merge is started for every activation of every incoming branch.
Discriminator: The discriminator is a point in a process that waits for one of the incoming branches to complete before activating the subsequent activity. From that moment on it waits for all remaining branches to complete and "ignores" them.
N-out-of-M Join: N-out-of-M Join is a point in a workflow process where M parallel paths converge into one. The subsequent activity should be activated once N paths have completed. Completion of all remaining paths should be ignored.
Structural Patterns
Arbitrary Cycles: A point in a workflow process where one or more activities can be done repeatedly.
Implicit Termination: A given subprocess should be terminated when there is nothing else to be done.
Patterns Involving Multiple Instances
MI without Synchronization: Within the context of a single process instance multiple instances of an activity can be created, i.e., there is a facility to spawn off new threads of control. There is no need to synchronize these threads.
MI with a priori design time knowledge: For one process instance an activity is enabled multiple times. The number of instances of a given activity for a given process instance is known at design time. Once all instances are completed some other activity needs to be started.
MI with a priori runtime knowledge: For one process instance an activity is enabled multiple times. The number of instances of a given activity is known at some stage during runtime, before the instances of that activity have to be created. Once all instances are completed some other activity needs to be started.
MI without a priori runtime knowledge: For one case an activity is enabled multiple times. Once all instances are completed some other activity needs to be started.
State Based Patterns
Deferred Choice: A point in the workflow process where one of several branches is chosen. Several alternatives are offered to the environment. Once the environment activates one of the branches the other alternative branches are withdrawn.
Interleaved Parallel Routing: A set of activities is executed in an arbitrary order: Each activity in the set is executed, the order is decided at run-time, and no two activities are executed at the same moment
Milestone: The enabling of an activity depends on the process instance being in a specified state.
Cancellation Patterns
Cancel Activity: An enabled activity is disabled, i.e. a thread waiting for the execution of an activity is removed.
Cancel Case: A workflow instance is removed completely.

performed in academy and industry. Section 5 concludes this paper.

2. WORKFLOW PATTERNS

According to [8] workflow specifications can be understood, in a broad sense, from a number of different perspectives. The *control-flow (or process) perspective*, for instance, describes activities and their execution ordering through different constructors. The *data perspective* adds operational business and processing data to the control flow perspective. The *resource (or organizational) perspective* provides an anchor to the organizational structure in the form of human and device roles responsible for executing activities. The *operational perspective* describes elementary tasks executed in activities which might map into underlying applications.

Thus, the control flow perspective provides an essential understanding into a workflow management system's effectiveness, while the data perspective rests on it, and the organizational and operational perspectives are ancillary. Therefore the workflow patterns described in [1] focus on the control-flow perspective. Work related to data and resource perspectives can be found in [9] and [10], respectively. Regarding the operational perspective, some focus is given to approaches to support tool integration, such as [11].

An overview of the workflow patterns, grouped by category, with a description adapted from [1], is presented in Table 1. Further details (description, synonyms, examples, problem statement, and potential implementation strategies) and illustrations can be obtained in [1] [2].

3. PATTERNFLOW

The workflow patterns were used as the set of control-flow requirements to be implemented by PatternFlow. Additional requirements were related to the data, resource, and operational perspectives. For the data perspective the management of attached documents should be supported. For the resource perspective it should be possible to organize users in groups (which could represent, for instance, their roles in the organization). Finally, for the operational perspective flexibility for implementing the tasks to be accomplished in each of the activities (the atomic elements of the process flow) and integration facilities to other applications should be provided.

The following subsections provide an overview of PatternFlow's architecture and main functionalities. Additionally an explanation of how the workflow patterns are supported is provided.

3.1 Architecture

The architecture was defined based on the WfMC reference model [12]. The main components and interfaces of this reference model are illustrated in Figure 1.

According to [13] some effort has been done on the establishment of generic interfaces, aiming to allow the exchange of information in a standardized way and thus enabling interoperability between different products. For instance, Wf-XML has been developed for engine-engine interactions, and has recently been used also for client-engine interactions. However, the generic interfaces are still a goal to be reached. The components of the reference model, on the other hand, provide an understanding of the elements that a workflow management system should have.

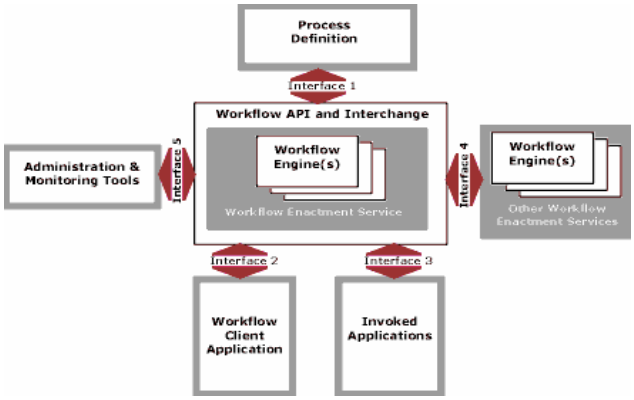


Figure 1. Components and interfaces of the WfMC reference model.

Based on the components of the reference model, the initial architecture was defined for PatternFlow as illustrated in Figure 2. It has two main modules: FlowServer and FlowDeveloper. Mapping this architecture to the reference model components, FlowDeveloper implements process definition and administration, whereas FlowServer implements the workflow engine and allows monitoring the process. The browser serves as the workflow client application of the model.

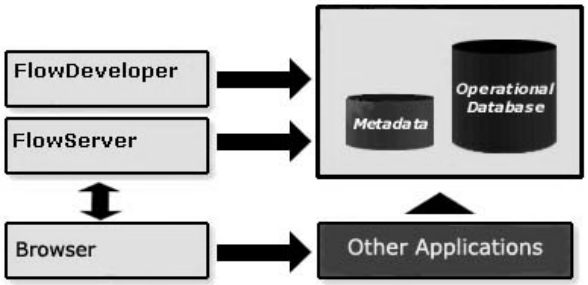


Figure 2. PatternFlow initial architecture.

Note that the architecture has two data repositories: one for the metadata and one for operational data. The metadata contains the description of the modeled processes and is used for their enactment. Therefore the metadata is used by both modules, FlowDeveloper (to define the metadata) and FlowServer (which uses the metadata to enact the process). The operational data contains application specific data used for implementing the tasks of a given activity. To implement those tasks PatternFlow allows linking activities to dynamic pages, or sets of dynamic pages organized in a menu. Moreover, it allows the integration of other external applications that might be helpful for accomplishing an activity by using the integration mechanism described in [11].

Figure 3 shows the entity relationship model of the metadata and how it usually relates to the operational data. Four of the tables shown in this model are generated each time a new process gets described. Those tables are related to the instances that are being enacted and start with the name of the process (wich is represented in the model as <INSTANCE>). The link to the operational data is usually done to an instance of the process (but depends on how the activities themselves are implemented). Further explanation regarding the metadata follows.

Basically users have access to groups which have access to activities which are part of a process description. Each activity has (1) rules to define the next activities to be enabled whenever those rules are satisfied (2) a form, describing a dynamic page or menu with a set of dynamic pages used for accomplishing the different tasks of the activity, and (3) a list of activities to be cancelled as soon as the activity is accomplished. An Instance is in a certain activity of a process and might have attached documents (besides the history of all activity transitions and operations on the attachments). If an instance of a process happens to be in more than one activity then a separate entry for the instance will exist for each activity in the instance table.

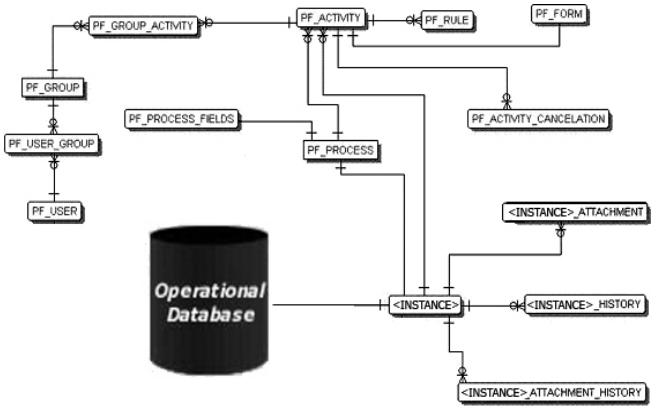


Figure 3. Metadata entity relationship model.

Additionally, besides the basic execution data (owner, arrival time, how many arrived, etc), an instance has a set of 20 control-flow fields that are used uniquely for the transition rules. Labels for those fields can be defined during the process description.

3.2 Main Functionalities

The main functionalities of PatternFlow can be divided in two categories: Process Description and Administration, and Process Enactment and Monitoring. Those two categories are handled respectively by FlowDeveloper and FlowServer. An overview of the functionalities in each of the categories is provided in the following two subsections.

3.2.1 Process Description and Administration

To illustrate how processes can be described in PatternFlow we prepared a small example process concerning the build or buy decision, which needs to be taken frequently in software organizations. This process is shown in Figure 4.

In short, once a manager needs to take a build or buy decision he launches a new analysis, than, two separate groups (called build evaluator and buy evaluator) start to work in parallel analyzing the build and the buy decision respectively. Afterwards the results of the two analyses are merged so that one of the managers can evaluate both, select one, and approve it or not. Actually, if he decides to not approve the selection doesn't really matter. The two final activities are just to indicate the status in which each process instance finishes the process.

Regarding the workflow patterns presented in section 2, this process uses the parallel split (also referred to as AND-Split, indicated by the upper AND), the synchronization (also referred

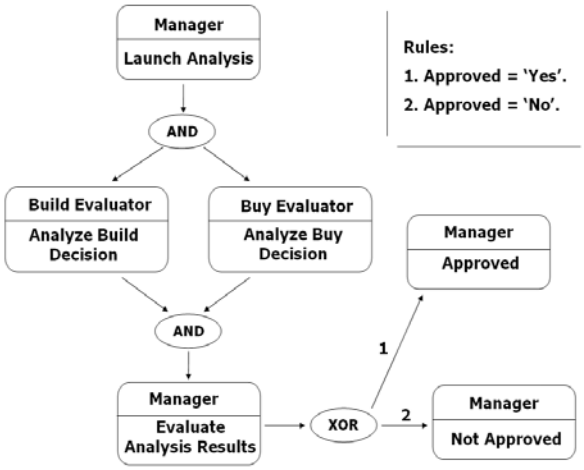


Figure 4. Example process: build or buy decision.

to as AND-Join, indicated by the lower AND), and the exclusive choice (indicated by the XOR and the deviation rules) patterns.

In PatternFlow this process can be defined using FlowDeveloper, which is linked to the metadata and allows creating and describing new processes. The main process description screen of this tool is shown in Figure 5.

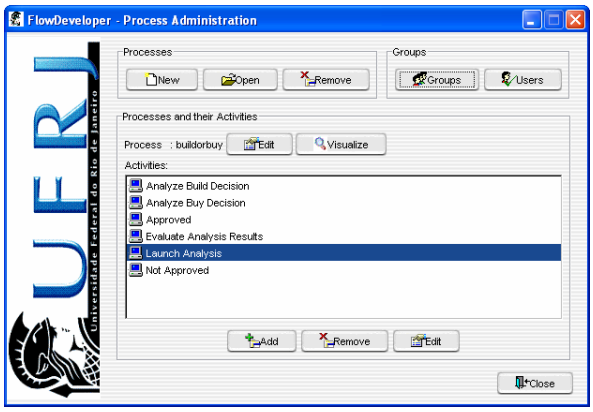


Figure 5. FlowDeveloper process description screen.

In this screen an empty build or buy process can be created and then the activities can be added to it. It also allows creating users and groups. The screen to edit a group is shown in Figure 6. Basically a group can be associated to the activities to which it has access and to the users which are part of it. Once a user becomes part of a group he automatically will be able to access activities related to the group through the browser (using a URL to access FlowServer, which will be described in the next subsection).

One of the main tasks of describing a process in PatternFlow is describing each of its activities. The screen to describe an activity is shown in Figure 7. It allows classifying the activity, redefining the groups associated to the activity, defining synchronization parameters (fan-in and discriminator), defining the link to the next activities (with the associated transition rules) and to the activities it cancels.

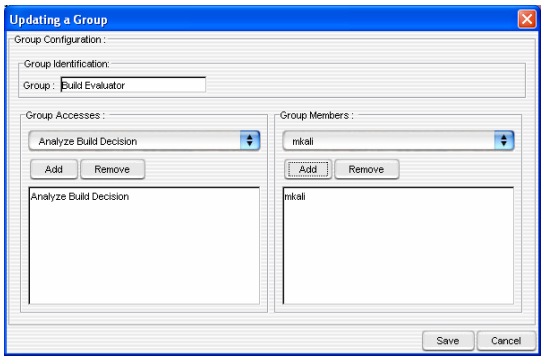


Figure 6. Updating a group in FlowDeveloper.

Moreover, a separate tab, shown in Figure 8, allows (1) defining the dynamic page (or menu with a set of dynamic pages) that is used for accomplishing the activity itself, (2) to determine if the activity enables ad-hoc process flow (selected by the user at runtime) to an activity or user, and (3) to select the activities for which the fan-in for an instance should be adjusted according to the number of destinations determined in runtime for the current activity (this is done to implement the 'Synchronizing Join' pattern, as described in the next section).

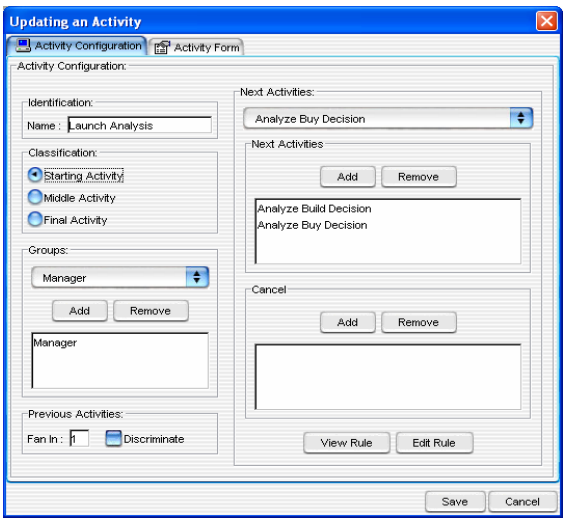


Figure 7. Describing an activity.

The activity described in Figure 7 and 8 is the 'Launch Analysis' of our example process. Note that it is classified as a starting activity (which means that in it a new instance of the process can be created). The group associated to it is the 'Manager' group. Since it does not synchronize any previous activities its fan-in is set to 1, and since it does not discriminate remaining arriving instances the discriminate checkbox is unchecked (refer to the discriminator pattern for further details on the discriminate checkbox). Its next activities are 'Analyze Build Decision' and 'Analyze Buy Decision', and the transition rule for both is 'always next activity' (no rule is defined).

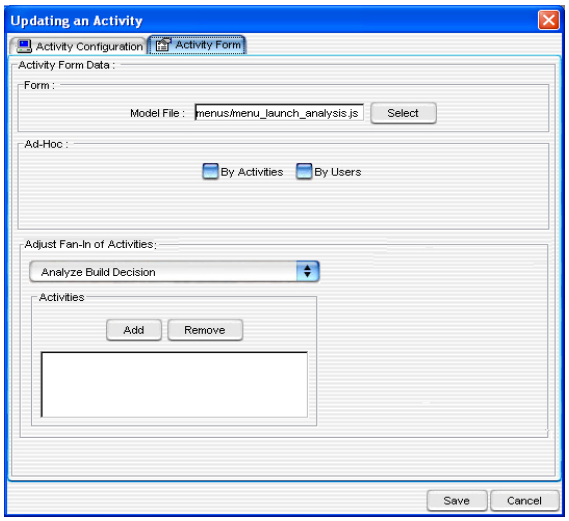


Figure 8. Defining a form for accomplishing the activity and if the activity allows an ad-hoc process flow.

Regarding the transition rules, they can be edited by selecting one of the next activities and then invoking 'Edit Rule'. The screen for defining a transition rule between two activities is shown in Figure 9. It allows mounting a logical expression regarding any field of the instance table, including the 20 fields uniquely dedicated to control-flow. Those 20 fields appear in the rule definition screen with the labels associated to them. The generated expression can also be directly edited. In Figure 9 the transition rule 'Approved = Yes' was entered to describe the transition from the 'Evaluate Analysis Results' to the 'Approved' activity of our example process.

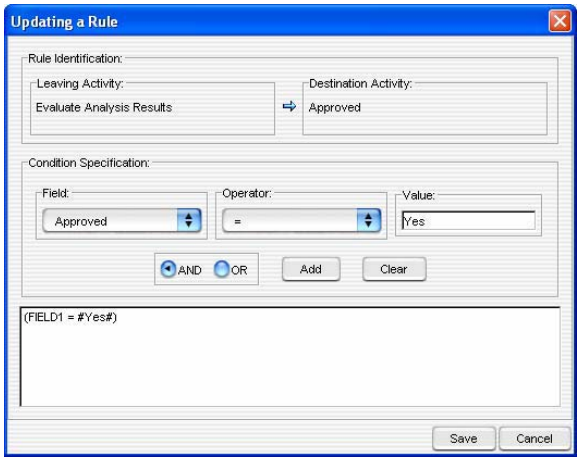


Figure 9. Defining activity transition rules.

To enable visualizing the process being described EnactPro [14] has been integrated. EnactPro is called as a dynamic link library and receives a XML description of the process, which PatternFlow generates according to EnactPro's XML Schema. The visualization of the example process in EnactPro is shown in Figure 10.

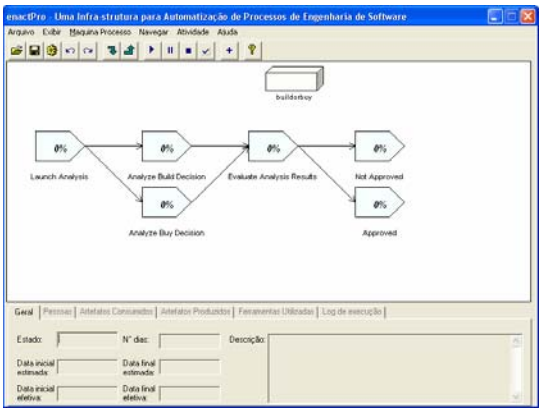


Figure 10. Showing the example process in EnactPro.

However, EnactPro was developed for its own purposes. Therefore it has several other features which are not part of the scope of this paper, but does not explicitly show the implemented workflow patterns nor the transition rules between activities. To illustrate the patterns and the transition rules in PatternFlow the visualization feature would have to be implemented considering the metadata of PatternFlow itself instead of the XML Schema of EnactPro.

3.2.2 Process Enactment and Monitoring

Once the example process has been described, support to the activities (the workflows atomic units of work) has to be implemented. Therefore the dynamic pages (or menus) specified for the activities need to be developed.

In the case of no need for dynamic processing of operational data a facility for building static models for the activities is provided (see [5] for further details on this facility) those models are interpreted in runtime and are able to update the basic control-flow fields of PatternFlow. However, in general, this is not the case and dynamic processing of operational data is needed.

Since FlowServer runs on a JEE container the dynamic pages specified have to be JSP's or servlets (unless they are processed on another container), which is a technological constraint to the default solution. In the case of using a menu several dynamic pages can be associated to an activity.

A facility for integrating other applications into the activity implementation is provided using the XMapper integration mechanism described in [11]. The usage of this mechanism in PatternFlow is shown in Figure 11. The activity has control over which applications (external tools) it provides to the user. The pre-condition for using the integration mechanism is that the tool should produce an XML artifact. XMapper then uses XML schemas and ontologies to generate transformation drivers (XSLT) to convert the XML of the tool into an XML that conforms to a schema that the activity implementation understands.

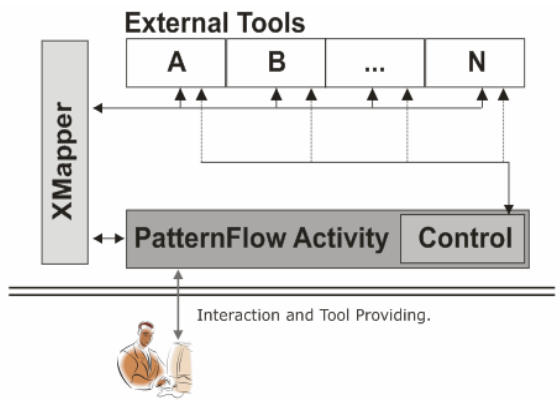


Figure 11. Using the integration mechanism.

Once the implementation of the support to each activity is done the process can be enacted and monitored using FlowServer.

FlowServer is available through an URL, displaying an interface such as shown in Figure 12. This static interface might be customized according to the process. In Figure 12, for instance we initialized the right part with an illustration of the build or buy decision process.

The process is then enacted in FlowServer following 4 main use cases, described below:

Entering the system. The user provides his username, password and the process he would like to log in (this last parameter might be fixed and provided in a hidden way, without the awareness of the end user). FlowServer then determines the groups to which the user has access and the activities of the process related to those groups. As a result a list of activities is shown to the user, as illustrated in Figure 13, where the activities related to the manager group of our example are displayed.

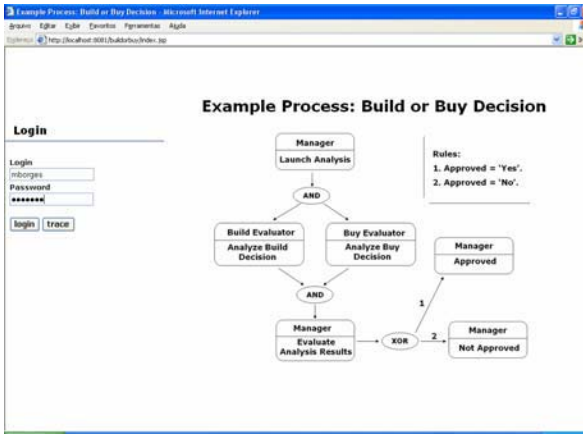


Figure 12. FlowServer process enactment and monitoring interface.

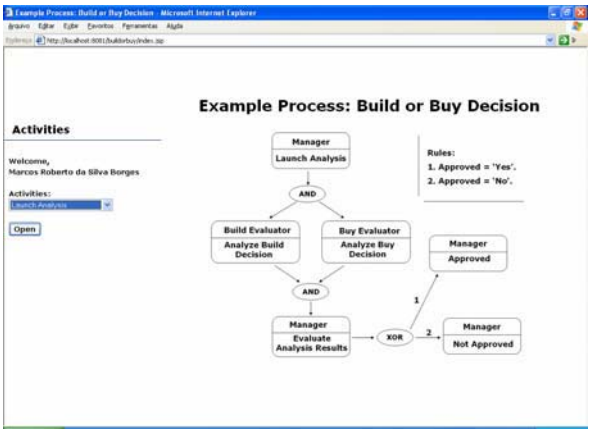


Figure 13. Choosing an activity.

Choosing an Activity. The user selects the activity in which he would like to perform his tasks. As a result FlowServer displays basic information of all the pending instances of the process situated in the activity. If the activity was classified as a starting activity FlowServer also enables creating new instances of the process. The result of selecting the activity 'Launch Analysis' of our example process is displayed in Figure 14, where, after selecting the activity a new instance of the process was created, to analyze whether to build or to buy a calendar component.

Choosing an Instance. The user selects the instance he would like to work on in the current activity. FlowServer then displays the implementation (representing the tasks of the activity) associated to the activity for the chosen instance through a static model, dynamic page, or menu. In Figure 15, a menu was associated, and the menu task shown is attaching the documents, which functionality is provided by FlowServer and does not require any implementation effort (just has to be associated to the menu, which can be done following the example of the sample menus). This functionality uploads the documents to the server and makes them available in all activities where the functionality is displayed in the menu.

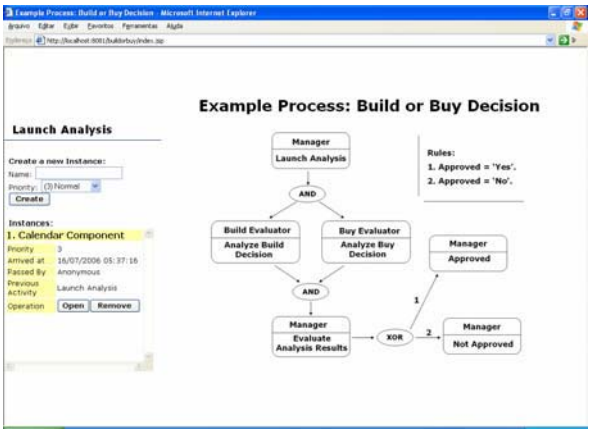


Figure 14. Choosing an instance.

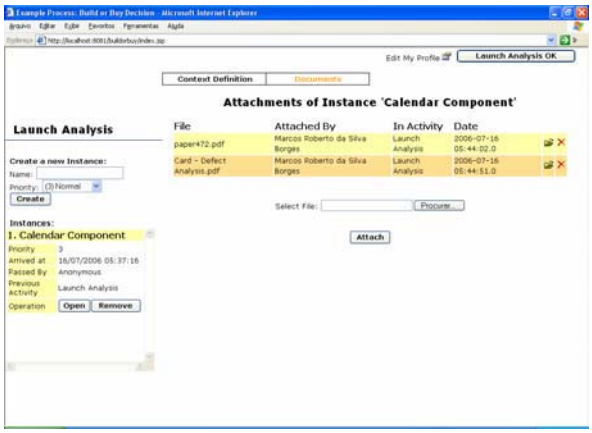


Figure 15. The activity implementation.

Accomplishing an Activity. Once all the tasks of an activity have been performed the user hits the upper right button for finishing the activity. At this moment FlowServer will process the transition rules for each of the next (destination) activities. Those rules are processed as an SQL statement on the instance table, considering the value of all its fields, including the additional control-flow fields. As a result, a message displays which transitions were done. Additionally an e-mail notification is sent to all members that have access to the destination activities. Figure 16 shows the message displayed after finishing the 'Launch Analysis' activity for the instance 'Calendar Component', were a transition to activities 'Analyze Buy Decision' and 'Analyze Build Decision' occurred.

After the transitions, if the number of arrivals of the instance at the destination is smaller than the fan-in of the activity it gets incremented and, in the case in which other arrivals of the same instance happened before in the activity, the instance data gets updated (merged). If the number of arrivals of the instance is already greater than or equal to the fan-in, another copy of the instance gets generated at the destination.

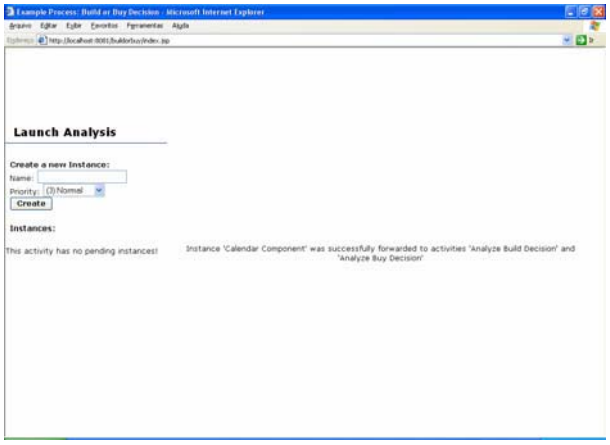


Figure 16. Accomplishing an activity.

Once the number of arrivals for a process instance is equal to the fan-in defined for the destination activity the instance is made available in this activity. In the case of our example the fan-in of both destination activities was defined as 1 and therefore the instance becomes immediately available, not waiting for any new arrival. Figure 17 shows the instance 'Calendar Component' available in activity 'Analyze Buy Decision'.

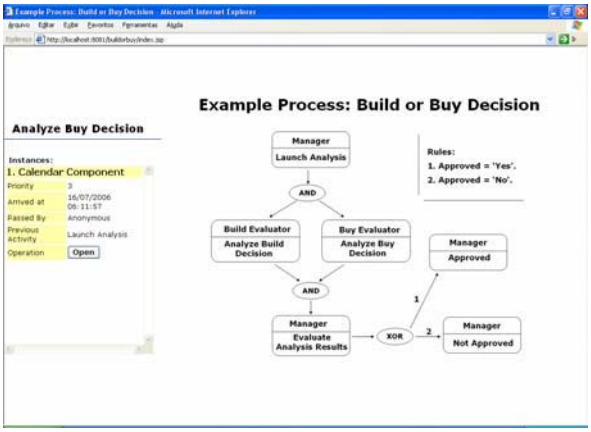


Figure 17. Instance 'Calendar Component' available in activity 'Analyze Buy Decision'.

The progress of an instance of the process can be monitored through FlowServer (hitting the 'Trace' button instead of logging into the system). Monitoring enables managers to take corrective actions in case of deviations from the expected enactment. Besides monitoring the instance itself, its attachments can also be monitored, showing when (and by whom) a given document was removed or added to the instance.

The monitoring instance functionality is shown in Figure 18, where the transition of our example instance from activity 'Launch Analysis' to activities 'Analyze Build Decision' and 'Analyze Buy Decision' is shown.

Provided the overview of the basic functionalities, the next section discusses how the workflow patterns were supported in PatternFlow.

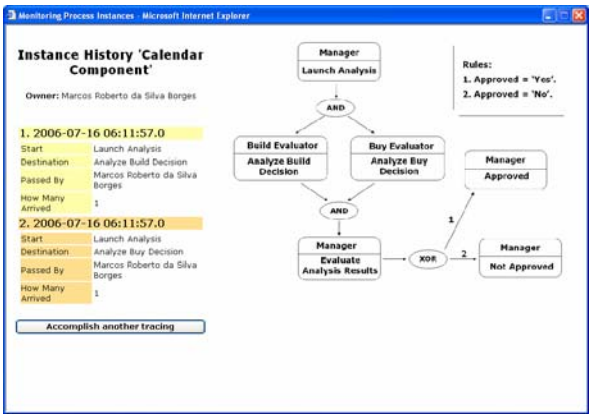


Figure 18. Monitoring the instances.

3.3 Supporting the Workflow Patterns

The description of how the workflow patterns were supported in PatternFlow is organized in the following subsections according to the categories of the patterns shown in Table 1.

3.3.1 Basic Control Flow

The basic control flow patterns can be easily described in FlowDeveloper and are supported by FlowServer. The 'Sequence' pattern is described by defining a next activity without a transition rule (always next activity) to a given activity. The 'Parallel Split' can be described by adding more than one next activities without transition rules to a given activity. The 'Exclusive Choice' can be implemented by defining more than one next activities with complementary transition rules (like in our example, where 'Approved = Yes' and 'Approved = No' were used).

For describing the 'Simple Merge' setting the fan-in of an activity to 1 is enough. For describing the 'Synchronization' pattern the fan-in of the activity should be defined in FlowDeveloper as being the amount of incoming activities to be synchronized.

3.3.2 Advanced Branching and Synchronization

Below follows a description of how the advanced branching and synchronization patterns can be described and enacted in PatternFlow.

The 'Multiple Choice' pattern can be described in FlowDeveloper by defining non complementary transition rules for the next activities of a given activity. For instance, defining a transition rule to one destination activity as being 'Approved = Yes' and to another as 'Decision = Buy'. In this case, if the instance got approved and the decision was to buy then it would be forwarded to both destinations.

The 'Multiple Merge' can be described by setting the fan-in of the destination activity to 1. Once an instance arrives, the number of arrivals will be equal to the fan-in and therefore, as soon as the next part of the same instance arrives a new copy of the instance will be generated.

The 'Synchronizing Merge' can be described by setting PatternFlow to update the fan-in for an instance in other activities according to the number of destinations found for that instance in the current activity in runtime. Thus, the fan-in for that instance will be matching the actual amount of branches taken, assuring that they will be synchronized. This pattern was not supported in the initial version of PatternFlow [5].

For describing the 'Discriminator' and 'N-out-of-M Join' patterns the discriminate checkbox has to be checked. Additionally, the fan-in should be set to 1 or to the number of branches to be considered, respectively.

3.3.3 Structural Patterns

The 'Arbitrary Cycles' pattern can be described without any restriction, since there is no limitation to which activities can be chosen as next activities. The 'Implicit Termination' is always present, since an activity in PatternFlow is considered an ending activity if there are no next activities defined for it.

3.3.4 Patterns Involving Multiple Instances

The 'MI without Synchronization' pattern can be described since an activity can enable copying the instance in it as many times as desired by setting a rule which defines it as its own next activity twice and setting its fan-in to 1. Thus, each time a new copy of the instance will be created (because arrivals will be greater than or equal to the fan-in).

The 'MI with a Priori Design Time Knowledge' can be described by associating the same activity several times as the next activity of its previous one (with the same transition rule). The result would be a parallel split of the instance. The next activities should synchronize the different parts of the instance (using the synchronization pattern).

Regarding the 'MI with a Priori Runtime Knowledge', it can be described by setting the same activity several times as the next activity, but with each transition associated to a different rule. Some hack might have to be implemented in order for the control-flow fields of the instance to reflect the desired runtime knowledge to be considered. Again, the next activities should synchronize the different generated parts of the instance (using the synchronizing merge pattern).

Finally, the 'MI without a Priori Known Runtime Knowledge' is currently not directly supported. This is because in the current solution the synchronization of the several instance parts would not be possible due to the unknown amount of branches to synchronize.

3.3.5 State Based Patterns

The 'Deferred Choice' pattern can be described by having the accomplishment of each of the activities of the possible branches canceling the remaining ones.

The 'Interleaved Parallel Routing' pattern is not supported yet, since semaphores are not considered in PatternFlow, only true parallelism is supported. It is important to mention that this pattern occurs rarely during the analysis phase of workflow projects.

Regarding the 'Milestone' pattern, since PatternFlow keeps the history of all activity transitions, it could be implemented as part of the support to the activity, setting the desired milestones into the control flow fields so that they could be used for describing the transition rules. No direct support is provided to this pattern yet.

3.3.6 Cancellation Patterns

The 'Cancel Activity' pattern can be described in FlowDeveloper by creating an activity that, when executed cancels another activity. The 'Cancel Case' pattern is present in FlowServer, which allows the owner of an instance to cancel it whenever he wishes.

Hence PatternFlow supports most of the workflow patterns (except 'MI without a Priori Known Runtime Knowledge', 'Interleaved Parallel Routing', and 'Milestone'). Moreover, it supports more patterns than each of the 15 workflow management systems evaluated in [1].

4. CASE STUDY: ISPIS

PatternFlow has been used, among others, to implement ISPIS [6], a framework for supporting the software inspection process. The following subsection provides an overview of this framework.

4.1 Overview

ISPIS implements the reorganization of the traditional inspection process described in [15], which introduces changes to reduce the cost and total time for the accomplishment of a particular type of

inspection, the inspections with asynchronous meetings performed by geographically distributed teams. Many of the remaining requirements for ISPIS were derived from knowledge acquired by empirical studies.

Basically the process consists of having a moderator planning the inspection, inspectors reviewing the artifact and generating discrepancy lists, the moderator filtering the resulting discrepancy lists for an asynchronous discrimination with the entire team to classify discrepancies as defects or false positives. Afterwards, the author takes the defect list and performs the rework on the artifact. Finally the moderator evaluates the quality of the inspection and of the corrected artifact and decides whether to perform another inspection or not.

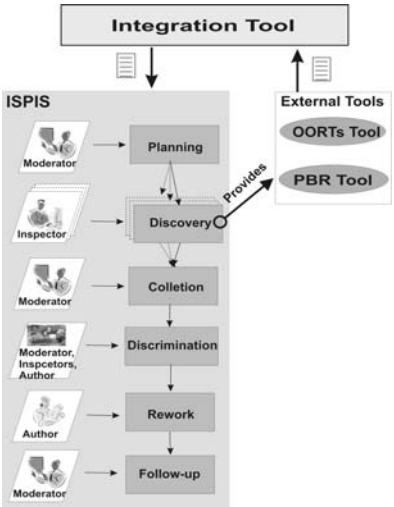


Figure 19. Inspection process and external tool integration mechanism.

Depending on the type of artifact and the technique to be used for the defect detection activity, the framework provides external tools. The discrepancy lists resulting from using those tools are then integrated into ISPIS using the integration mechanism described in Section 3. The inspection process and the integration mechanism are illustrated in Figure 19.

Besides describing the process in PatternFlow, operational data regarding software inspections (discrepancies, discrepancy lists, etc) was modeled to serve as a basis for implementing the support to each of the processes activities. The entity relationship model describing such operational data and how it relates to PatternFlow's metadata is shown in Figure 20.

Implementing ISPIS was an opportunity to improve the initial version of PatternFlow [5]. Support to more of the workflow patterns was implemented and many other improvements (like, for instance, multi-language support, allowing to associate menus to activities, displaying a summary of the pending instances of an activity instead of just the name, and enhancing the interfaces).

Having a real industry process supported by PatternFlow enabled us to perform some experimentation in academy and industry. A short description of those evaluations and the feedback obtained are described in the following subsections.

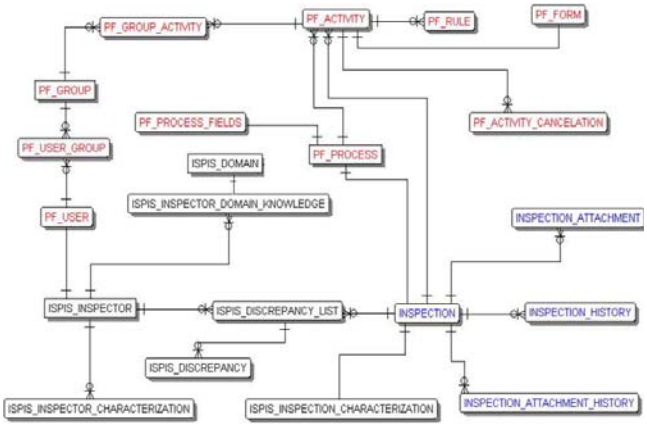


Figure 20. ISPIS's operational data and PatternFlow's metadata.

4.2 Evaluations in Academy and Industry

ISPIS has been carefully evaluated in the academic context where an experimental study and a case study have been conducted [6]. The experimental study was related to specific support provided by ISPIS to software inspections. The case study, on the other hand was related to process enactment.

The case study was carried out by conducting a real requirements document inspection. Subjects (12) were divided in two inspection teams of 5 members each, one author, and one moderator. In order to get feedback regarding the integration mechanism one of the groups used an external tool for the individual discovery activity. Feedback was gathered through a follow up questionnaire.

Both treatments (with and without using external tools) successfully reached the end of the inspection process and participants' feedback was positive. The main claims were regarding specific activity support implementations and not the process enactment itself.

Results of those evaluation studies showed promising results. However, the validity of those evaluations was scoped to the academic context.

Aiming to introduce ISPIS into the industrial context of Sakonnet Technology, a geographically distributed software organization, a comparison of those evaluations and the complete methodology for introducing software processes described in [16] was performed. As a result a set of two additional case studies to be conducted was derived: 'Use in real life cycle' and 'Use in industry'. The first of those case studies addresses how the process integrates into a real development lifecycle. The second one evaluates if the process fits into an industrial setting. Those case studies were conducted at Sakonnet Technology [7], by teams distributed between Rio de Janeiro, London, and New York.

Again, results of the studies showed positive feedback regarding process enactment and only suggestions for improving specific activity implementations of ISPIS were provided. Figure 21 shows a successfully finished inspection at Sakonnet Technology, in which 33 defects where found on a requirements document.

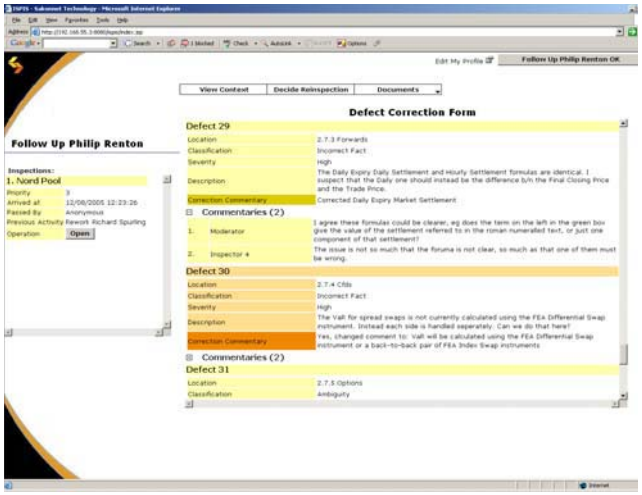


Figure 21. Follow-up activity of a real inspection using ISPIS at Sakonnet Technology.

4.3 Other Experiences

Currently PatternFlow is being used to enact several software inspections accomplished through the ISPIS framework, which has been used in real projects undertaken by the COPPETEC foundation.

One recent usage that deserves special attention was a simultaneous enactment of four inspections, with 5 team members each, having 20 people using ISPIS at the same time. These inspections happened in the context of a contest promoted by the IHC (human computer interfaces) Symposium 2006 to evaluate the interface of the JEMS (Journal and Event Management System) system. Fortunately PatternFlow handled the coordination of the process instances appropriately and the teams were able to complete their inspections without major problems.

5. CONCLUSIONS

The establishment of patterns to systematically address workflow requirements is important, since it should be possible for a modeled process to be described and enacted by commercial workflow management systems. The initiative of establishing those patterns was taken [1, 2, 3, and 4] and helped to understand in which extent these requirements are addressed in current workflow management systems.

However, most of the existing workflow solutions support only a relatively small subset of the more advanced patterns, and even the basic patterns are not supported by a great amount of them [1]. Since many of those patterns recur quite frequently during the analysis phase of workflow projects [1], some business processes, depending on their complexity, can not be described nor enacted by any of those workflow management systems.

Based on this scenario we implemented PatternFlow, a framework for describing and enacting business processes. The control-flow requirements of PatternFlow were derived from the analysis of the documented workflow patterns and most of them (except 'MI without a Priori Known Runtime Knowledge', 'Interleaved Parallel Routing', and 'Milestone') are supported in the resulting implementation.

Regarding process description, it is important to mention that the patterns can be described using the FlowDeveloper module, but not through visual modeling. To lower this impact PatternFlow was integrated into EnactPro [14] to allow the visualization of the process being modeled. Regarding enactment, the patterns are supported by the FlowServer module.

Over the years PatternFlow evolved and has been used, for instance, to implement ISPIS [6], a framework for supporting the software inspection process. Having a real industry process described in PatternFlow allowed us to perform evaluations in academy and industry [7]. Feedback of those evaluations and other usage experiences suggest the suitability of PatternFlow for enacting processes in real industrial environments.

6. ACKNOWLEDGMENTS

The authors would like to thank for the contributions of the experimental software engineering (ESE) team of COPPE/UFRJ (http://www.cos.ufrj.br/~ese) and the CHORD group of NCE/UFRJ (http://chord.nce.ufrj.br). Thanks also to CNPq for financial support.

7. REFERENCES

[1] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., Workflow Patterns, Journal on Distributed and Parallel Databases, 14, 3 (July 2003), 5-51.

[2] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., Workflow Patterns Home Page, http://www.workflowpatterns.com, last accessed July 15th 2006.

[3] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., Advanced Workflow Patterns, 7th International Conference on Cooperative Information Systems (CoopIS 2000), volume 1901 of Leture Notes in Computer Science, pages 18-29. Springer-Verlag, Berlin, 2000.

[4] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., Workflow Patterns, BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, 2000.

[5] Kalinowski, M., "PatternFlow: um software de workflow para a implementação e execução de processos de negócio padronizados", Bachelor in Computer Science Conclusion Project, DCC/UFRJ, 2001.

[6] Kalinowski, M., Travassos, G.H., A Computational Framework for Supporting Software Inspections, 19th IEEE International Conference on Automated Software Engineering, pp. 46-55, Linz, Austria, 2004.

[7] Kalinowski, M., Travassos, G.H., Software Technologies: The Use of Experimentation to Introduce ISPIS - a Software Inspection Framework - Into the Industry, 2nd Experimental Software Engineering Latin American Workshop (ESELAW 2005), 2005.

[8] Jablonski, S., Bussler, C., Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thompson Press, 1996.

[9] Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P., Workflow Data Patterns, QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane, 2004.

[10] Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P., Workflow Resource Patterns, BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven, 2004.

[11] Spinola, R.O., Kalinowski, M., Travassos, G. H., "Uma Infra-Estrutura para Integração de Ferramentas CASE", In: Proc. of the XVIII Brazilian Symposium on Software Engineering, pp. 147-162, Brasilia - DF, Brazil, 2004.

[12] WFMC, Workflow Management Coalition, http://www.wfmc.org/, last accessed July 15th 2006.

[13] Hollingsworth. D., The Workflow Reference Model 10 Years On, Chapter of the Workflow Handbook 2004, available at http://www.wfmc.org/, 2004.

[14] Mafra, S.N., Barros, M.O., Travassos G.H., "enactPro: Automatizando Processos de Software", In: XVIII Brazilian Symposium on Software Engineering – Tool Session, Brazil, 2004

[15] Sauer, C., Jeffery, D.R., Land, L., Yetton, P., "The Effectiveness of Software Development Technical Review: A Behaviorally Motivated Program of Research", IEEE Transactions on Software Engineering, 26 (1): 1-14, Jan. 2000.

[16] Shull, F., Carver, J., Travassos, G.H., "An Empirical Methodology for Introducing Software Processes." In: Proc. of European Software Engineering Conference, Vienna, Austria, pp. 288-296, 2001.

Selecting Anomalous Traces from Workflow Event Logs

Fábio Bezerra
IC – UNICAMP
Caixa Postal 6176
Campinas, SP, Brazil
fbezerra@ic.unicamp.br

Jacques Wainer
IC – UNICAMP
Caixa Postal 6176
Campinas, SP, Brazil
wainer@ic.unicamp.br

RESUMO

Este trabalho apresenta um modelo de detecção de *traces* anômalos gerados por um sistema gerenciador de processos. A detecção é baseada no *custo de inclusão* de um *trace* sobre um modelo construído através de um algoritmo de mineração de workflow que utiliza um conjunto de heurísticas. O cálculo do custo de inclusão é realizado como segue: *(i)* um *trace* do log é removido; *(ii)* um modelo de workflow baseado em um subconjunto dos *traces* restantes é descoberto; e *(iii)* o *trace* removido é adicionado sobre o modelo minerado. O custo de inclusão representa o tamanho da modificação de um modelo após a inclusão de um *trace* no mesmo. Se um *trace* modifica o modelo, então ele será selecionado como anômalo, senão indica uma instância do modelo. Este artigo explica o método proposto, alguns exemplos, bem como nossa avaliação do método.

ABSTRACT

This work shows a model to detect a set of anomalous traces in a log generated by a business process management system. The detection of an anomalous trace is based on the *inclusion cost* of a trace in a workflow model constructed by workflow mining based on rewriting rules. The inclusion cost is evaluated as follows: *(i)* one trace is removed from log; *(ii)* a workflow model is mined from a random subset of remaining traces of log; *(iii)* the removed trace is merged over model. The *inclusion cost* is the amount of modification of a model after the inclusion of trace on it. If a trace produces a modification on the model, it will be selected as an abnormal or anomalous trace, otherwise it is an instance of the mined model. This paper explains how the inclusion cost works and the proposed model to select the anomalous traces. Moreover, we comment the results of experiments and some limitations.

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 98-106.

Keywords

Workflow Mining, Anomaly Detection, Information Security

1. INTRODUCTION AND MOTIVATION

Workflow mining [9, 3, 5, 4] is a technique used to reconstruct a workflow model from a set of traces in a log. The techniques used to mine a workflow trace log can be used to discover how people work and to support business process modeling. For example, an ERP system might not enforce a way of work in some procedures. Then a log of these transactions could be used to model an usage pattern of such system, that is, organizational procedures not strongly defined can be mined to suggest a process model [8]. Another use of workflow mining techniques is to construct social networks as presented in [10].

However, current research in workflow mining usually assumes that the log is in some way complete (under some appropriate definition of completeness [9]) so that the "correct" workflow model can be inferred from the log. The current workflow research also assumes that the log is correct, that is, all traces in the log were generated from the workflow model which will be discovered by the process mining. This work challenges this last assumption.

In this work we are interested in applying workflow mining techniques to discover anomalies or outliers in the log. These anomalies may be exceptional executions, or insecure executions of the workflow, or even fraud attempts.

Anomaly detection has been the target of most research in the field of data mining, since many surprising, rare events are of interest in security, surveillance, epidemiology, fraud detection and so on. For example, disease outbreak detection has been proposed by detecting anomalies in the event logs of emergency visits [1], or the retail data for pharmacies [6]. In [2] the association rules are cited as an important tool to find tendencies in databases. But the huge amount of rules obtained even for small databases implies serious drawbacks to this approach. So, they proposed an approach that discovers anomalous association rules, that can be used to analyze deviations from normal behavior.

In this work we adopt the following rationale to detect anomalies: the traces that are not anomalies when mined should generate a workflow model, and trying to fit the anomaly into this model will probably require a lot of structural changes. The anomaly is then the trace whose inclusion produces a modification to the model generated by the other traces of the log. In order for this rationale to work, we need a workflow mining algorithm that is incremental, that is, which makes it possible to generate a new workflow model by including a trace into an "old" workflow model. We use the rewriting rules proposed in [11].

This paper is organized as follow: in Section 2 we present the workflow mining method used to implement the anomaly detection model. A prototype of this model is commented. In Section 3 we present our proposal model for anomaly detection, the *inclusion cost*. Moreover, we present some examples that illustrate the proposed model and methodology used to validate it. In Section 4 we discuss some related works and in Section 5 we point at some conclusions and future works.

2. WORKFLOW MINING

In this section we present the rewriting method of workflow mining, how it works and some issues of its implementation. Finally, an extended model is presented.

2.1 Rewriting Method

The rewriting method is a technique proposed in [11] for process mining. This method is considered an incremental method, because a known model is remodeled or rewritten based on a given workflow trace that isn't an instance of the original model. Then, trace by trace a model could be incrementally modified.

A model is rewritten during its merging with a trace. If the trace is an instance of the model, nothing is done. Otherwise the model is modified. The rewritten model is constructed based on application of a set of rules. Below we list the notation that we will follow and a revised set of rules that was originally presented in [11].

Notation:

- $\alpha, \beta, \gamma, \delta$ represent a sequence of activities or structural blocks (AND, OR, LOOP);
- $\alpha\beta$ represents the sequence of α followed by β;
- $\alpha + \beta$ represents the OR-split/OR-join block of branches α and β;
- $\alpha \parallel \beta$ represents the AND-split/AND-join block of branches α and β;
- $\alpha \top \beta$ represents a loop in which α is the main branch, and β is the alternative path;
- $\oplus$ is the amalgamation operator;
- $M \oplus \alpha$ denotes the amalgamation of a model M (first argument) and a trace α (second argument);
- if α is a model expression, then α' is a sequence of activities that can be generated from α

Structural rules:

S1 $\alpha \oplus \alpha' \Rightarrow \alpha$ If the new trace can be generated from the model, there is no need to change the model.

S2 $\alpha\beta \oplus \gamma\delta \Rightarrow (\alpha \oplus \gamma)(\beta \oplus \delta)$ Segmentation of two traces for an alternative amalgamation.

S3 $\alpha + \beta \oplus \gamma \Rightarrow (\alpha \oplus \gamma) + \beta$ An alternative amalgamation inside an OR block;

S4 $\alpha \top \beta \oplus \gamma \Rightarrow (\alpha \oplus \gamma) \top \beta$ An alternative amalgamation inside a LOOP block;

S5 $\alpha \parallel \beta \oplus \gamma\delta \Rightarrow (\alpha \oplus \gamma) \parallel (\beta \oplus \delta)$ An alternative amalgamation inside an AND block;

Introduction rules:

Or-I $\alpha \oplus \beta \Rightarrow \alpha + \beta$ If two traces can't be merged, a new model is constructed with an OR operator.

And-I $\alpha\beta \oplus \beta'\alpha' \Rightarrow \alpha \parallel \beta$ Rule used to generate an AND operator. The trace $\beta'\alpha'$ is an inverted instance of $\alpha\beta$.

And2-I $(\alpha \parallel \beta)\gamma \oplus \alpha'\gamma'\beta' \Rightarrow (\alpha\gamma) \parallel \beta$ Rule used to generate an AND operator.

And3-I $\gamma(\alpha \parallel \beta) \oplus \alpha'\gamma'\beta' \Rightarrow \alpha \parallel (\gamma\beta)$ Rule used to generate an AND operator.

Loop-I $\alpha \oplus \alpha'\beta\alpha' \Rightarrow \alpha \top \beta$ Rule used to generate a LOOP operator.

Below we illustrate how the application of rules on a given set of traces $T = \{abcde, acbe, abce\}$ works. The first two traces are merged producing a model that is merged with the third trace.

$abcde \oplus acbe \Rightarrow$
$(a \oplus a)(bcde \oplus cbe)$; S2 rule

$a(bcde \oplus cbe) \Rightarrow$
$a(bcd \oplus cb)(e \oplus e)$; S2 rule

$a(bcd \oplus cb)e \Rightarrow$
$a[(bc \oplus cb)(d \oplus \epsilon)]e$; S2 rule

$a[(bc \oplus cb)(d \oplus \epsilon)]e \Rightarrow$
$a(b \parallel c)(d \oplus \epsilon)e$; And-I rule

$a(b \parallel c)[(d \oplus \epsilon)]e \Rightarrow$
$a(b \parallel c)(d + \epsilon)e$; Or-I rule

$a(b \parallel c)(d + \epsilon)e$; mined model

$a(b \parallel c)(d + \epsilon)e \oplus abce \Rightarrow$
$(a \oplus a)[(b \parallel c)(d + \epsilon)e \oplus bce]$; S2 rule

$a[(b \parallel c)(d + \epsilon)e \oplus bce] \Rightarrow$
$a[(b \parallel c) \oplus bc][(d + \epsilon)e \oplus e]$; S2 rule

$a(b \parallel c)[(d + \epsilon)e \oplus e] \Rightarrow$
$a(b \parallel c)[(d + \epsilon) \oplus \epsilon](e \oplus e)$; S2 rule

$a(b \parallel c)(d + \epsilon)e$; S1 rule

$a(b \parallel c)(d + \epsilon)e$; mined model

The execution trace presented above is one of many possible execution traces of process mining. For each two merged trace we can have many models as a solution. The model obtained in the first step when merged with a third trace can generate another set of models, and so on.

This property of process mining with rewriting rules has some limitations. The *first* limitation is execution time, because many different models could be generated and we are interested in one of them. The *second* one is the dependency of the solution model on the order of the traces. The execution trace shown below illustrates this limitation. Since we know that the given solution is consistent with all traces used in process mining, then, both solutions $a(b \parallel c)(d+\epsilon)e$ and $a[(b \parallel c) + cbd]e$ could instantiate the traces of set T above. The *third* limitation is related to the previous ones. It is concerned with the selection criteria that is employed to choose an appropriate rule among the rules that can be applied to rewrite the model.

$abce \oplus acbe \Rightarrow a(b \parallel c)e$
$a(b \parallel c)e \oplus abcde \Rightarrow a[(b \parallel c) + cbd]e$

$a[(b \parallel c) + cbd]e$; mined model

2.2 Extending the Rewriting Model

One drawback of mining based on rewriting rules is that many models could be constructed from a log, because during the amalgamation of a model and a trace, different sequences of rules could be applied. So, many models are generated from the log. To solve this limitation, we developed a function that makes a global searching in the mined model space. Figure 1 illustrates an example of global search. The selection of the smallest model reduces the search space and is a more deterministic approach to mine a process model.

Among all derived models, the function we propose selects the smallest one. A model M is smaller than a model M' if M has less vertices than M'. This approach reduces the search space and improves the execution time of the rewriting method. However, it is possible that two mined models have the same number of vertices. This scenario points out that this algorithm is still a non-deterministic solution, because in this case, it randomly chooses a model.

3. ANOMALY DETECTION

In this section we present our technique to detect anomalous traces based on inclusion cost. First we present how inclusion cost is evaluated. Then, we explain the proposed model to select all anomalous traces. Finally, we discuss some experiments we have conducted.

3.1 Inclusion Cost

The technique for detecting anomalous traces we are proposing is based on the evaluation of the application cost of a trace over a known workflow model. The *inclusion cost* is the difference of size between the two models. The size of a model is the number of vertices that it has. Figure 2 illustrates how a mining works. The figure shows three models generated when mining a set of traces $T = \{abcd, acbd, abce\}$. The first trace could be an instance of a sequential model whose size is four, as depicted. After merging of trace *acbd* with the sequential model, a bigger model is generated with size six. Then a third trace is merged, generating a third model with size nine.

This simple example illustrates how the inclusion of a trace produces a modification in the model. Each generated model has to be able to instantiate the same set of traces that constructed it. Therefore, the trace *acbd* could not be instantiated by sequential model, then a modification was needed. The same applies to the third trace.

A trace that is an instance of a workflow model does not modify it, such a trace can not be an anomalous, because the inclusion cost will be zero. This principle is used by the workflow mining process based on rewriting rules. During the mining, a model will not be modified if it is merged with an instance trace. For example, if we try to introduce the trace *acbe* nothing will be done on the last model of Figure 2, because *acbe* is an instance of the model.

3.2 Selecting Anomalous Traces

We select the anomalous traces after calculating the inclusion cost of all different traces of a log. The traces that have an inclusion cost higher than zero are selected. The important consideration is about the model used to calculate the inclusion cost after merging. Therefore, the mined model is constructed based on a set of random chosen traces. We assume that anomalous traces are rare and they are anomalous because they are not an instance of the workflow model that generated the "normal" traces. This is not the only possible definition of anomaly, but is the one we use in this paper. Our proposed method considers this assumption when selecting the traces of a log to construct the model.

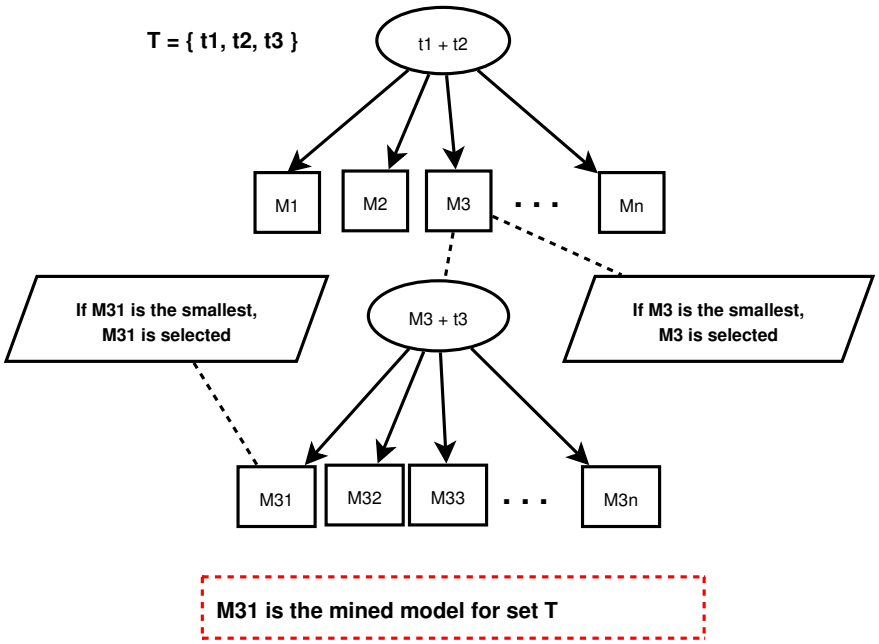


Figure 1: Global search for traces in set T.

Figure 3 illustrates how a log is preprocessed during execution of proposed method. L is the original log, D is a set of different traces and R is a set of randomly selected traces. The random traces are used to mine a model that will be merged with a trace of D. For each trace of D, a set R of random traces is generated, a model is mined, and an inclusion cost for the mined model is calculated. A trace is selected if it has a non-zero inclusion cost. For example, when a potential anomalous trace T_a is selected from set D, we remove the first occurrence of T_a in L, and a R' set is generated. Then a model M is mined from R' and merged with trace T_a. If T_a is an anomalous trace, it will produce a modification over model M and it will be selected as anomalous. We present a pseudocode for this algorithm in Figure 4.

Assuming that anomalous traces are rare events, when an anomalous traces is removed from the log during the anomaly detecting process, it is improbable that the same trace occurs again in the log. Then, the model will be constructed without instances of the anomalous trace, which will be modified after the amalgamation. However, if two or more occurrences of an anomalous trace exist in the log, this anomalous trace might not be selected as anomalous. We can minimize this problem defining a small selection factor (for example, less than 50%).

3.3 Experiments and Methodology

We are interested in discovering an anomalous trace in a log, but an anomaly is difficult to be defined, mainly in business processes where a complete definition of model is not well known. Hence, in order to validate the presented method, we developed three models and populated a log with all possible traces that these suggested models can instance, repeating occurrences. We then added some traces which are not instances of the created models. Using this approach we can guarantee that an occurrence of a different trace of the generated possible traces is necessarily an anomalous one.

Figures 5, 6 and 7 represent the models used. The first example, Model 01, has seven activities with an AND-Split/Join block and an OR-Split/Join block, but only six activities can be instantiated. The second example, Model 02, has five activities, but only four activities can be instantiated. The third example, Model 03, has nine activities and is a bigger model with nested blocks in it. The possible traces of these models are represented in Table 1.

We got the possible traces presented in Table 1 and populate three files with them, generating three logs for experiments. We generated the files repeating the traces, since repetition increases the chance of "normal" traces to be selected during the composition of set R (see Figure 3). Then, three traces that are not instances of Models 01, 02 and 03 were created and added to the log files in a random position. Table 2 illustrates the anomalous traces created.

We divided the experiments in three groups. In the *first group* we created three log files with possible traces of Table 1, repeating occurrences. Then, in the *second group* we have got the log files of the first group and added the anomalous traces of Table 2. Finally, on the *third group* we have got the

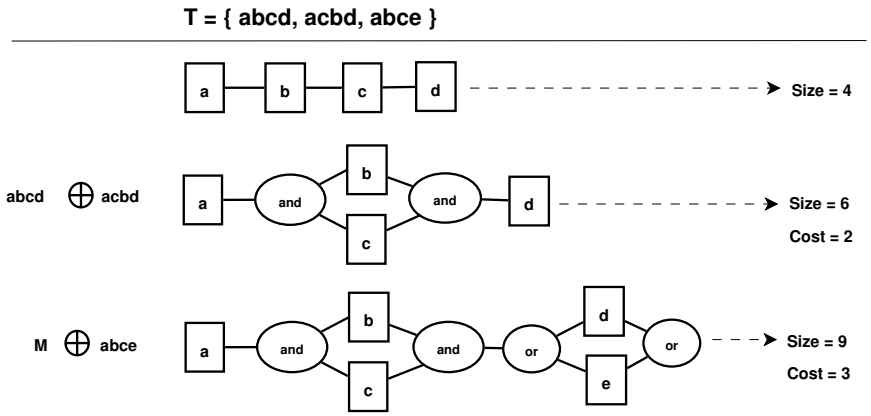


Figure 2: Mining example of a set of traces T.

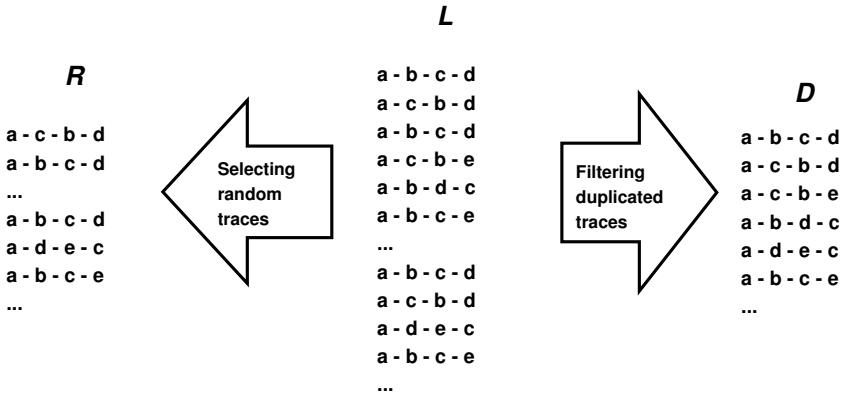


Figure 3: Preprocessing the log. Random choosing is repeated for each different trace.

log files of the second group and added the anomalous traces of Table 2 again, generating files with repeated instances of anomalous traces. The anomalous traces included in the log files represent at most 4% of all traces. Therefore, we had for each model: *(i)* a log with repeated occurrences of possible traces, *(ii)* a log with repeated occurrences of possible traces and one occurrence of each suggested anomalous trace, and *(iii)* a log with repeated occurrences of possible traces and two occurrences of each suggested anomalous trace. After that, we applied the proposed method on the logs.

3.4 Results of experiments

For each set (possible traces of Models 01, 02 and 03; Table 1) we created a log file with repeated instances in about 100 traces. We applied our method to these logs (first group) and none anomalous trace was detected. This behavior was expected because, even removing the first occurrence of trace T from log L, the resulting set of traces ($L' = L - T$) has many instances of the same trace T. Therefore, trace T still has a high chance to be selected during composition of set R, composed of traces randomically chosen (see Figure 3). But, if trace T is not selected during composition of set R, the remaining "normal" traces can still mine a model which can generate T as an instance. Figure 8 illustrates how a log without all the possible traces can mine a model that will be equal to the model if mined with all possible traces. For example, the first incomplete log of figure mines an *OR block* when the two first traces are merged. Then, the resulting model is merged with third trace, generating an *AND block*, which is equal to the depicted model. On the other hand, each pair of traces mine a block of the prescribed model.

Others experiments considered logs with anomalous traces, as shown in Table 2. First we added only one occurrence of each trace from Table 2 in the log (second group). Then, we added two repeated occurrence of each anomalous trace in the log (third group). We got excellent results when applied the method with logs without repeated instances of anomalous traces, because we detected all anomalous occurrences inserted in the log and none "normal" trace was detected as anomalous. We call a "normal" detected trace as *false*

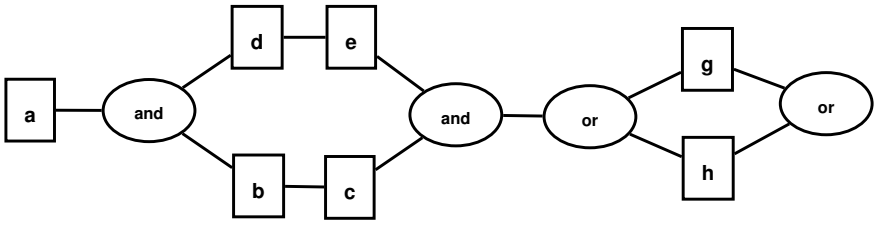


Figure 5: Model 01: a process model with seven activities.

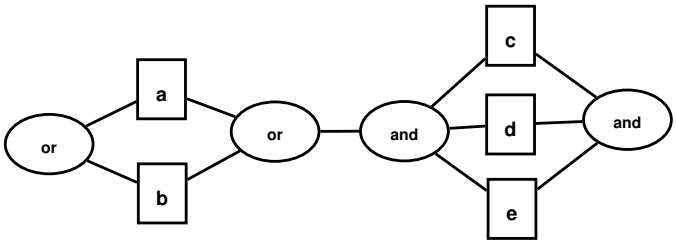


Figure 6: Model 02: a process model with five activities.

```
L      := Set of traces from log;
//Set of different traces
D      := filter(L);
//Selection factor of traces
Factor := 50%;
//Set of anomalous traces, initially empty
A      := [ ];
for each trace T of D do
   L'  := L - T;
   R   := random(Factor, L');
   //Mining a model M1
   M1  := mine(R);
   CM1 := size(M1);
   //Merging M1 and trace T
   M2  := mine([M1, T]);
   CM2 := size(M2);
   if (CM2 - CM1) > 0 then
      A := A + T;
   end if;
end for;
```

Figure 4: A pseudocode for anomaly detection algorithm.

positive. However, even without any occurrences of false positives in our experiments with complete and correct logs (that is, logs with only the possible traces), we know that false positives could occur even in these cases.

Other tests were made on the logs with repeated instances of anomalous traces. In these experiments the results were not as good as the tests made on the logs without repetition of anomalous traces, because some anomalous traces were not detected. We call these undetected anomalous traces as *false negatives* and they occur because when T is removed from log L, there is a 50% chance (which is the selection factor used) that the same remaining anomalous trace T in the log is selected. In the experiment with the third group, our method did not report any false positive. Table 3 summarizes the experiments.

Model 01	Model 02	Model 03
a, d, e, b, c, g	a, e, c, d	a, d, c, e, g, h
a, b, d, e, c, g	a, e, d, c	a, c, d, e, g, h
a, b, c, d, e, g	a, c, d, e	a, c, e, d, g, h
a, d, b, e, c, g	a, c, e, d	a, d, c, e, i
a, d, b, c, e, g	a, d, e, c	a, c, d, e, i
a, b, d, c, e, g	a, d, c, e	a, c, e, d, i
a, d, e, b, c, h	b, e, c, d	b, f, g, h
a, b, d, e, c, h	b, e, d, c	b, f, i
a, b, c, d, e, h	b, c, d, e	
a, d, b, e, c, h	b, c, e, d	
a, d, b, c, e, h	b, d, e, c	
a, b, d, c, e, h	b, d, c, e	

Table 1: Possible traces of models 01, 02 and 03.

4. RELATED WORKS

The application of process mining techniques on event log to audit security violations is present in [8]. In this work, the event log is called audit trail and the process mining technique is used to discovery a security pattern. Two logs are used, one to audit and the other to apply the mining process. This separation guarantees that the second log only has traces with normal behavior. That is, the log contains traces that are not considered security violations. Process

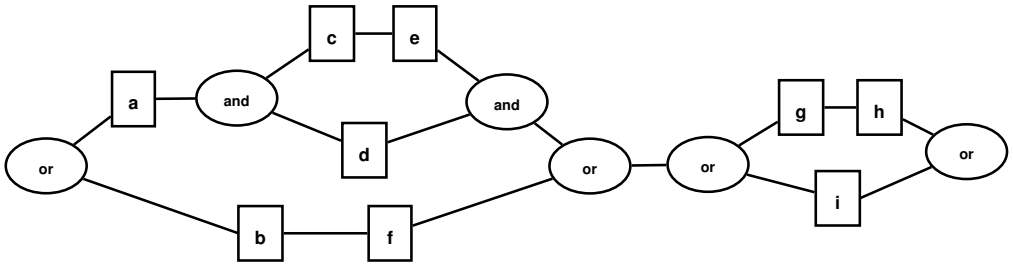


Figure 7: Model 03: a process model with nine activities.

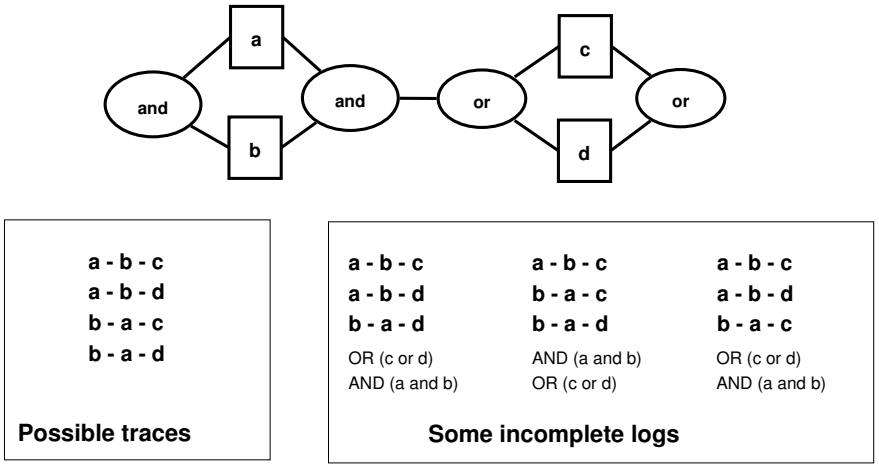


Figure 8: Mining a model with incomplete set of traces.

Model 01	Model 02	Model 03
a, b, c, g, h	a, b, c, d	a, c, e, g, h, i
a, b, c, d, h, g	a, c, d, b	a, c, d, i, g, h
a, g, h, b, c, d	b, c, e	a, b, c, f, i

Table 2: Anomalous traces for models 01, 02 and 03.

mining is then applied, generating an acceptable model (security pattern) which will be used to detect abnormal traces in the audit log.

During the detection, if the abnormal traces do not represent a security violation, they are included in the mining log and the security pattern evolves. However, comparing with our work this is a clearing limitation, because a normal log is known, and therefore a security pattern is not difficult to find. In our work, we do not known if a log has abnormal traces, so the inclusion cost method is used to select a trace if it exists.

Another approach presented in [8] is the conformance test applied to the audit log. It is similar to our S1 rule, but they do not consider the whole process model, but some fragments that represents a causal relation in the process. This causal relation is obtained from the process model mined with the normal log. However, it is not clear how the causal relation is generated. Again, we think that the main limitation of this approach is the use of a log only with normal traces.

In [7] the author presents two interesting methods to compare the alignment of a prescribed process model and a discovered process model. The first approach is delta analysis and the second is the conformance testing. The basic idea of delta analysis is to measure how different a mined process is from the descriptive one, while the conformance testing is based on the analysis of an event log with a process model. This work is not directly related to ours, because the author does not present a solution to detect an abnormal or an anomalous trace in a log. However, he presents an interesting measure of process alignment, that could be an alternative method to detect anomalies in a log similarly inclusion cost.

The inclusion cost approach tries to determine the difference of two processes based on vertices counting, that is, the number of activities and OR/AND blocks. While our approach seems simple, it does not consider the existence of a descrip-

Experiment	Model	Repetition	False negatives
1st	01	No	None
2nd			None
3rd			None
1st	01	Yes	{a, b, c, g, h}, {a, b, c, d, h, g}
2nd			{a, b, c, g, h}, {a, b, c, d, h, g}, {a, g, h, b, c, d}
3rd			{a, g, h, b, c, d}
1st	02	No	None
2nd			None
3rd			None
1st	02	Yes	{a, c, d, b}, {b, c, e}
2nd			{b, c, e}
3rd			{a, c, d, b}
1st	03	No	None
2nd			None
3rd			None
1st	03	Yes	{a, c, e, g, h, i}
2nd			{a, c, e, g, h, i}, {a, c, d, i, g, h}
3rd			{a, c, e, g, h, i}

Table 3: Results of experiments. Logs with anomalous traces.

tive or prescribed model, so if a trace makes a big change (model size) in a mined model (until now, unknown model), this trace is possibly a anomalous one. Moreover, the mining method we used, rewriting rules, tries to rearrange a known model to accommodate the new trace, generating a new rewritten model.

5. CONCLUSION AND FUTURE WORKS

Process mining techniques are focused on discovering models to support process definitions and discovering a social network inside an organization. These techniques can help managers to understand how people work. Workflow trace log is an interesting source of data that can be used to discover normal patterns. However, little attention has been paid to the application of process mining techniques on discovery of anomalous traces. Among the applications of this approach we can cite fraud detections and security violation attempts.

In this work we have shown how workflow mining based on rewriting rules can be used to detect anomalous traces. We presented an algorithm that marks a trace as anomalous if this trace, when merged with the model mined with rewriting rules, has a non-zero inclusion cost. The mined model is based on a subset of randomly chosen traces. The inclusion cost is higher than zero if the merged trace forces a modification on the original model.

The inclusion cost method demonstrated in this work proved to be a good solution, mainly when applied on the log without repeated occurrences of anomalous traces. However, an investigation on bigger models has to be done. That is, models with many activities, loops and deeper nested blocks. Another investigation is about a selection factor smaller than 50%. In these case we expected to diminish the occurrence of false negatives, but to present some false positives.

Our experiments have not presented results with false positives, however we think that false positives occurrences do not represent a big limitation. The human intervention is always needed whose could correctly discard a false detection.

The proposed method still has limitations when applied to a log with repeated instances of anomalous traces. In this case it is possible that an anomalous trace can not be selected (a false negative). However, assuming that anomalous traces are uncommon, successive executions could select this trace, and a union of results could correctly select all anomalous occurrences. Another approach is to define a threshold cost, so the selected anomalous traces will be the ones that have an inclusion cost which is higher than this threshold.

6. REFERENCES

[1] D. K. Agarwal. An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In *ICDM*, pages 26–33, 2005.

[2] M.-A. Balderas, F. Berzal, J. C. Cubero, E. Eisman, and N. Marn. Discovering hidden association rules. In *Proceedings of the KDD 2005 Workshop on Data Mining Methods for Anomaly Detection*, August 2005.

[3] A. de Medeiros, W. van der Aalst, and A. Weijters. Workflow mining: Current status and future

directions. In R. Meersman, Z. Tari, and D. Schmidt, editors, *On The Move to Meaningful Internet Systems*, volume 2888 of *LNCS*, 2003.

[4] M. Hammori, J. Herbst, and N. Kleiner. Interactive workflow mining - requirements, concepts and implementation. *Data Knowl. Eng.*, 56(1):41–63, 2006.

[5] L. Maruster, W. M. P. van der Aalst, T. Weijters, A. van den Bosch, and W. Daelemans. Automated discovery of workflow models from hospital data. In Krse, B., Rijke, M., Schreiber, G., and Someren, M., editors, *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pages 183–190, 2001.

[6] R. Sabhnani, D. Neill, and A. Moore. Detecting anomalous patterns in pharmacy retail data. In *Proceedings of the KDD 2005 Workshop on Data Mining Methods for Anomaly Detection*, August 2005.

[7] W. M. P. van der Aalst. Business alignment: using process mining as a tool for delta analysis and conformance testing. *Requir. Eng.*, 10(3):198–211, 2005.

[8] W. M. P. van der Aalst and A. K. A. de Medeiros. Process mining and security: Detecting anomalous process executions and checking process conformance. *Electr. Notes Theor. Comput. Sci.*, 121:3–21, 2005.

[9] W. M. P. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1128–1142, 2004.

[10] W. M. van der Aalst Minseok Song. Mining social networks: Uncovering interaction patterns in business processes. In J. Desel, B. Pernici, and M. Weske, editors, *Business Process Management: Second International Conference*, volume 3080 of *Lecture Notes in Computer Science*, pages pp. 244 – 260, 2004.

[11] J. Wainer, K. Kim, and C. A. Ellis. A Workflow Mining Method Through Model Rewriting. In H. Fuks, S. Lukosch, and A. C. Salgado, editors, *Groupware: Design, Implementation, and Use: 11th International Workshop*, volume 3706, pages p. 184–19, Porto de Galinhas, Brazil, Setembro 2005. CRIWG 2005.

Distribuição de tarefas em sistemas de workflow por meio de seleção induzida de recursos

Autran Macêdo[1], Ilmério R. Silva[1], Rogério S. Silva[2], João N. Souza[1]
[1]Faculdade de Computação
[2]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Uberlândia
{autran, ilmerio}@facom.ufu.br
rogerio@pos.facom.ufu.br
nunes@ufu.br

RESUMO

Análise de ligações (*Link Analysis*) é uma técnica para classificação de documentos (páginas) da *web* e HITS é um algoritmo que implementa essa técnica. A classificação é realizada considerando a "relevância" das páginas. A relevância de uma página p é determinada pelos seus *links* de entrada (outras páginas que apontam para p) e de saída (páginas que são apontadas por p). Este artigo apresenta a aplicação de análise de *links*/HITS no contexto de *workflow*. A aplicação visa apresentar uma proposta de distribuição de tarefas considerando que recursos apontam para tarefas, as quais esses possuem competência para executá-las. São apresentadas algumas abordagens para essa aplicação, que são comparadas com trabalhos correlatos, em termos quantitativos e qualitativos. O aspecto quantitativo está relacionado com o tempo de execução do *workflow*. O aspecto qualitativo está relacionado com o resultado (produto ou serviço) obtido com a execução do mesmo. Os experimentos realizados mostram que a utilização do HITS na distribuição de tarefas em *workflow* representa ganhos da ordem de 25% em termos quantitativos e resultados ligeiramente superiores em termos qualitativos, quando comparados com trabalhos similares.

ABSTRACT

Link Analysis is a technique for ranking web pages and HITS is an algorithm that implements this technique. The ranking is computed by considering the pages relevance. The relevance of page p is determined by its input links (other pages that point to p) and output links (pages that p points to). This work presents the use of Link Analysis in Workflow as a new approach (as far as we know) for task scheduling. The association between Link Analysis and Workflow is accomplished by considering that resources point to workflow tasks that they are able to run. We present some implementations of this approach and we compare them against some related works. This comparison takes into account quantitative aspect (workflow execution time) and qualitative aspect (product or service delivered according to its requirements). The experiments have been shown gains around 25% in terms of quantitative aspect and results slightly superior in term of qualitative aspect against similar works.

Palavras-chave:

workflow, distribuição de tarefas, escalonamento, análise de ligações, hits, *hub*, autoridade,

Keywords

workflow, task scheduling, link analysis, hits, hub, authority

1. INTRODUÇÃO

Distribuição de tarefas é uma atividade importante em sistemas de workflow. Essa importância está relacionada à necessidade de se assegurar que uma dada tarefa seja executada pelo recurso correto no tempo devido. A distribuição de tarefas é uma das atividades realizadas pelos sistemas de gerência de workflow (SGWf). Em geral, os SGWfs adotam a política FIFO de distribuição associada a uma das estratégias abaixo[14]:

Push - o SGWf seleciona um recurso para executar uma dada tarefa;

Pull - o SGWf apresenta uma dada tarefa a um conjunto de recursos; um deles seleciona a tarefa para execução.

A estratégia *Push* é mais restritiva do que a *Pull*, mas há maior controle sobre execução da tarefa, em termos qualitativos. Considerações inversas podem ser feitas para a estratégia *Pull*. Alguns trabalhos [11, 13, 15], no entanto,

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 107-117.

têm demonstrado que a adoção de políticas de distribuição mais sofisticadas promovem maior eficiência no sistema de workflow como um todo.

Assim sendo, este trabalho apresenta a aplicação da técnica de Análise de Ligações (*L*ink Analysis [1]), cuja origem remonta o problema de citações em Biblioteconomia. Essa técnica tem sido aplicada para classificar documentos (páginas) da *web* [2, 3, 12] conforme suas respectivas relevâncias.

A relevância de uma página é calculada levando-se em conta as ligações de entrada (*in-links*) e de saída (*out-links*) de cada página. Uma página que é apontada por várias outras possui ligações de entrada. Uma página que aponta para várias outras possui ligações de saída. Esse relacionamento de apontamento (ligação) entre páginas *web* produz um esquema de propagação em dois níveis denominado **relacionamento mutuamente reforçado**. Isto é, se uma página é apontada por muitas outras ela é considerada uma **autoridade**, e se uma página aponta para muitas autoridades ela é denominada de ***hub***. A figura 1 ilustra esse relacionamento.

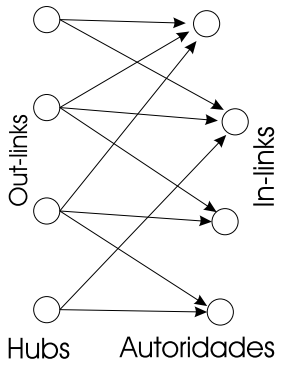


Figura 1: Ligações de entrada e de saída entre páginas da *web*.

O relacionamento mutuamente reforçado da Análise de Ligações pode ser percebido em *workflow* com relação a tarefas e recursos. As tarefas "são apontadas" por recursos cujas competências os habilitam à execução das mesmas. Por sua vez, os recursos "apontam" para tarefas as quais eles têm competência para executar. O algortimo HITS [10], que implementa a técnica de Análise de Ligações, inspira este trabalho para distribuir tarefas em ambientes de *workflow*. Até onde se estendeu esta pesquisa, trata-se de uma abordagem inédita.

2. TRABALHOS RELACIONADOS

Tramontina [13] propõe uma técnica baseada em algoritmos genéticos (*Guess and Solve*) para tratar problemas de incerteza quanto (1) aos tempos de execução de cada atividade (tarefa) e (2) às rotas de execução de cada caso (processo). O foco desse trabalho é planejar uma possível seqüência de tarefas antes do processo ser instaciado. O foco do presente trabalho está na distribuição de tarefas de processos já instanciados.

Kumar, et. al. [11] visualizam a distribuição de tarefas conforme perspectivas de qualidade e desempenho, tentando criar um balanço entre elas. Nesse trabalho os autores relacionam a segurança com a qualidade das associações, considerando que as tarefas são atribuídas sempre aos recursos altamente capacitados. São apresentados mecanismos para estimar a aptidão e a taxa de ocupação dos recursos, a carga de trabalho do sistema e a qualidade de execução dos casos. Os mecanismos de distribuição propostos – baseados na estratégia *Pull* – distribuem as tarefas apresentando-as para os recursos mais qualificados de acordo com limiares de adequação. Esses mecanismos possibilitam uma melhor sintonia entre qualidade e desempenho.

O trabalho de Veloso e Macêdo [15] apresenta uma proposta de distribução de tarefas tendo como base a aptidão dos recursos. Neste contexto são apresentados três mecanismos baseados na estratégia *Push* para a distibuição de tarefas. Esse trababalho apresentou resultados que demonstraram ganhos consideráveis relativos ao tempo de execução e à qualidade em comparação ao trabalho de Kumar et al [11].

Este trabalho utiliza os critérios de qualidade e desempenho de Kumar, et. al., mas a abordagem baseada no HITS o torna completamente diferente. Além disso, o trabalho desses utilizam a estratégia *Pull* enquanto o presente trabalho utiliza a estratégia *Push*, com melhores resultados. Quanto ao trabalho de Veloso e Macêdo, o presente trabalho implementa uma abordagem diferente (HITS) cujos resultados são melhores do que esse.

3. O ALGORITMO HITS

O algoritmo *HITS* (Hypertext Induced Topic Selection) [10] foi proposto originalmente para classificar páginas (documentos) da *web*. É possível classificar os documentos considerando-se o grau de autoridade. Isto é, no topo do *ranking* estão os documentos com maiores graus de *autoridade*. Por outro lado, também é possível classificar documentos considerando-se o grau de *hub*. Neste caso os documentos que estão no topo possuem ligações para os documentos com maior grau de *autoridade*.

O HITS opera sobre um grafo dirigido $G = (N, AR)$, que representa a ligação (*in-links* e *out-links*) entre os documentos. N corresponde ao conjunto de vértices do grafo (documentos) e AR o conjunto de arcos (ligações). O algoritmo opera sobre dois vetores: $H[n]$ corresponde ao grau de *hub* do documento $n \in N$; e $A[n]$ corresponde ao grau de *autoridade* do documento $n \in N$. A figura 2 ilustra o algoritmo HITS proposto originalmente.

Inicialmente x recebe o número de elementos de N (linha 1) e cada posição dos vetores H e A é iniciada com 1 (linhas

1 $x \leftarrow | N |$
2 $H[1..x] \leftarrow 1$
3 $A[1..x] \leftarrow 1$
4 **enquanto** H e A não convergirem
5 $\forall n \in N$
$$A[n] = \sum_{(m,n) \in AR} H[m]$$
6 $\forall n \in N$
$$H[n] = \sum_{(n,m) \in AR} A[m]$$
7 Normalize H e A
8 **fim_enquanto**

Figura 2: Algoritmo HITS

2–3). Em seguida, em cada iteração do laço (linhas 4–8), o grau de autoridade (linha 5) e de *hub* (linha 6) são calculados. $A[n]$ é igual à soma dos graus de *hub* de todas os documentos que apontam para n. $H[n]$ é igual à soma dos graus de autoridade de todos os documentos para os quais n aponta. Cada iteração termina com uma normalização, tal que $\sum_{i=1}^{x} H[i] = 1$ e $\sum_{i=1}^{x} A[i] = 1$. O laço termina quando H e A convergirem, isto é quando os valores desses vetores não tiverem mudança significativa entre a iteração corrente e a imediatamente anterior. Kleinberg [10] prova essa convergência e experimentalmente mostra que a mesma ocorre em torno da vigésima iteração.

4. APLICAÇÃO DO HITS EM WORKFLOW

As ligações (*in-links* e *out-links*), no domínio do HITS, ocorrem entre objetos do mesmo tipo: documento (página) da *web*. As ligações em *workflow* ocorrem entre objetos de tipos diferentes: recurso e tarefa. Assim sendo, foi adotado o seguinte critério para as ligações: as tarefas possuem *in-links* e os recursos possuem *out-links*. Deste modo, os recursos apontam para as tarefas as quais esses possuem competência para executá-las. Conseqüentemente, as tarefas são apontadas pelos recursos que têm competência para executá-las. Assim sendo, os recursos possuem graus de *hub* e as tarefas graus de autoridade. A figura 3 ilustra essa relação.

O relacionamento mutuamente reforçado do HITS avalia um *hub* como "bom" se este aponta para "boas" autoridades. Por outro lado, uma "boa" autoridade é aquela que é apontada por "bons" *hubs*. Paralelamente, em workflow, "bons" *hubs* são os recursos que têm competência para executar "boas" autoridades, e "boas" autoridades são as tarefas que podem ser executadas por "bons" *hubs*. Uma síntese do mapeamento entre HITS e *workflow* é apresentada na tabela 1.

4.1 Algoritmo AbsAllocHITS

Seguindo a linha dos algoritmos propostos por [1, 4, 5, 6, 8, 7, 9, 10, 17], AbsAllocHITS é um algoritmo para a classificação de recursos e tarefas que utiliza os conceitos de *hub* e autoridade.

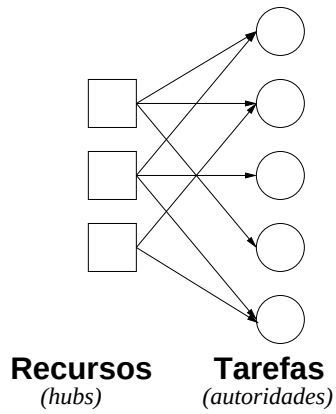


Figura 3: Grafo de associação de recursos à tarefas.

Tabela 1: Mapeamento entre conceitos de Workflow e HITS.

HITS	Workflow
Hub	Recurso
Autoridade	Tarefa
in-links	Tarefa é apontada por recursos com competência para executá-la
out-links	Recurso aponta para as tarefas as quais tem competência para executá-las

O algoritmo AbsAllocHITS opera sobre um grafo dirigido $G = (N, AR)$, que representa a ligação entre os recursos e as tarefas. N corresponde ao conjunto de vértices do grafo, composto pelo conjunto de recursos R e de tarefas T, portanto, $N = R \cup T$. AR corresponde ao conjunto de arcos (r, t), tal que $r \in R$ e $t \in T$. O algoritmo opera sobre dois vetores H e A que representam, respectivamente, os graus de *hub* e de autoridade.

O AbsAllocHITS insere o conceito de "valor de alocação absoluta" (VAB) [11] que é calculado a partir de alguns parâmetros relacionados a recursos e tarefas:

disponibilidade - grau de ocupação de um recurso;

adequação (r, t) - grau que indica quão adequado é o recurso r para executar a tarefa t;

urgência (t) - grau que indica a urgência da tarefa t;

conformidade (r, t) - grau que indica a violação de regras quando o recurso r executa t.

O VAB é utilizado para dar peso aos arcos (r, t), conforme

proposta similar de uma versão do HITS com peso [19]. O algoritmo AbsAllocHITS utiliza a função $fvab : R \times T \rightarrow [0,1]$, tal que $fvab(r,t)$ retorna o VAB de r em relação a t. A figura 4 ilustra o algoritmo AbsAllocHITS proposto.

1. $x \leftarrow \mid R \mid$
2. $y \leftarrow \mid T \mid$
3. $H[1..x] \leftarrow 1$
4. $A[1..y] \leftarrow 1$
5. **enquanto** H e A não convergirem
6. $\forall t \in T$
   $$A[t] = \sum_{\forall r \mid (r,t) \in AR} fvab(r,t).H[r]$$
7. $\forall r \in R$
   $$H[r] = \sum_{\forall t \mid (r,t) \in AR} fvab(r,t).A[t]$$
8. Normalize H e A
9. **fim_enquanto**
10. Classifique H e A em ordem crescente

Figura 4: Algoritmo AbsAllocHITS

No algoritmo AbsAllocHITS, inicialmente x recebe o número de elementos de R (linha 1) e y recebe o número de elementos de T (linha 2). Cada posição dos vetores H e A é iniciada com 1 (linhas 2–3). Em seguida, em cada iteração do laço (linhas 5–9), o grau de autoridade (linha 5) e de *hub* (linha 6) são calculados. $A[t]$ é igual à soma do produto entre $fvab(r,t)$ e $H[r]$, para todo arco (r,t). $H[r]$ é igual à soma do produto entre $fvab(r,t)$ e $A[t]$, para todo arco (r,t). O laço termina quando H e A convergirem, isto é quando os valores desses vetores não tiverem mudança significativa entre a iteração corrente e a imediatamente anterior. Ao final da execução, os vetores de *hub* e autoridade são classificados em ordem crescente.

A ordenação é em ordem crescente porque se quer deixar a alocação dos "bons" *hubs* por último. "Bons" *hubs* são os recursos versáteis, isto é aqueles que têm competência para executar diferentes tarefas. Deixando-os por último, aumenta-se a chance de se encontrar um recurso apropriado para uma tarefa.

5. EXPERIMENTOS

Nesta seção são apresentados o cenário da proposta, a estratégia para realização dos experimentos, e o ambiente de simulação adotado.

5.1 Cenário

Um cenário de workflow foi definido com a intenção de prover meios para a comparação deste trabalho com os trabalhos de [11, 13, 15]. Para garantir a conformidade dos experimentos de [11, 15], reproduziu-se o mesmo cenário utilizado. As definições e os parâmetros adotados também seguem a linha destes trabalhos. Este cenário é apresentado na figura 5.

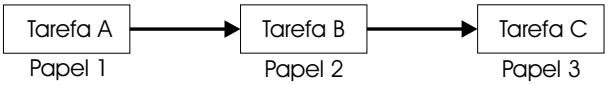


Figura 5: Processo de negócio utilizado nos experimentos.

O processo de negócio adotado neste cenário possui 3 tarefas em sequência (Tarefa A, Tarefa B e Tarefa C). Uma tarefa é uma unidade de trabalho que pode ser realizada de forma manual, semi-automática ou automática. As tarefas estão associadas respectivamente aos papéis (Papel 1, Papel 2 e Papel 3). Um papel é um modelo abstrato que reflete os aspectos do sistema, e que agrupa recursos afins. Um recurso é um participante concreto de um sistema de workflow, ele é incumbido de executar as tarefas. A cada papel estão associados um conjunto de recursos (cinco neste experimento, vide tabela 3). Desta forma, um recurso estará apto a executar as tarefas que estão associadas aos papéis os quais ele pertence.

O tempo da tarefa é o tempo definido para que a tarefa seja executada. Já o tempo do recurso é o tempo que o recurso demanda para executar uma determinada tarefa. As tarefas foram definidas com um tempo variando em torno de um padrão de execução de $10ut$(unidades de tempo). Porém para que o ambiente de simulação se aproximasse mais da realidade, foram gerados tempos distintos para os recursos.

Os experimentos realizados consistiram em gerar automaticamente várias instâncias do processo de negócio apresentado. Cada instância de execução do processo é gerada levando em consideração diferentes cargas de trabalho. A carga de trabalho (*workload*) é considerado um aspecto crucial no desempenho de um sistema de workflow [13]. Esta é definida como o número de tarefas que estão prontas para serem executadas no sistema. Foram definidos alguns parâmetros que afetam diretamente a carga de trabalho das instâncias para evitar que os experimentos se tornassem homogêneos (Veja tabela 2). A diversificação dos tempos das tarefas e dos tempos dos recursos tornam o ambiente de experimentação mais heterogêneo.

Tabela 2: Parâmetros de Simulação

	baixa	média	alta
Carga média de sistema	0,2	0,6	1,0
Variabilidade de tarefas	0,2	0,8	1,4
Variabilidade de recursos	0,2	0,8	1,4

O parâmetro "carga média de sistema" mede a proporção com que as tarefas ficam prontas para serem executadas. Ele determina o fluxo de novas tarefas que chegam ao sistema. Por exemplo, uma carga média de sistema com patamar $0,2$ indica que somente 20% da capacidade do sistema será utilizada. A diversificação da carga de trabalho das instâncias de um processo workflow é obtido a partir de uma

distribuição de *Poisson* com $\lambda = (l.n)/tm$, onde l é a carga média de sistema, n é o numero de recursos e tm é o tempo médio das tarefas.

O parâmetro variabilidade de tarefas determina como será a geração da amostra populacional diversificada de tarefas. As tarefas geradas terão um tempo de processamento variando em torno do valor padrão($10ut$). Para a tarefa t_j, a média de tempo de processamento $\overline{p_j}$ é gerada com base em uma distribuição normal com média tm (tempo médio das tarefas) e desvio padrão *varTarefa*. Assim, $\overline{p_j} \approx N(tm, (tm \times varTarefa)^2)$. Por exemplo, considerando uma variabilidade de tarefas $0,2$, teremos um tempo de processamento variando no intervalo $[8ut, 12ut]$.

A variabilidade dos recursos indica o percentual de variação dos tempos que os recursos consomem para executar as respectivas tarefas. O cálculo do tempo de processamento P_{ij} de uma tarefa t_i por um recurso r_j, ocorre de maneira similar à variação de tarefas. P_{ij} é calculado através de uma distribuição normal onde $p_{ij} \approx N(\overline{p_j}, (\overline{p_j} \times varRecurso)^2)$. Por exemplo, caso uma tarefa tenha seu tempo de execução definido como $10ut$ e a variabilidade de recursos seja $0,2$, significa que o recurso executará aquela tarefa com um tempo que irá variar no intervalo $[8ut, 12ut]$.

Conforme os parâmetros apresentados na tabela 2, o sistema tem a capacidade de gerar 540 condições experimentais, ou seja, para cada uma das 27 combinações possíveis da tabela são geradas aletoriamente 20 instâncias de problemas. Cada estratégia de distribuição será avaliada sob estas condições.

Outros parâmetros para a realização da simulação são a *disponibilidade* de cada recurso, a *urgência* de uma tarefa, a *conformidade* de um recurso para com uma tarefa, e a *adequação* de um recurso para executar uma determinada tarefa. Em todos os experimentos, os valores de *disponibilidade*, *urgência*, e *conformidade* seguem [11], onde são fixados em 1,0, 0,8 e 1,0, respectivamente. Os valores de *adequação* são apresentados na tabela 3. Esta tabela mostra também o mapeamento *papel* x *tarefa*, associando os recursos pertencentes aos papéis às suas respectivas tarefas. Um pequeno contingente de recursos (cinco recursos associados a cada papel) foi adotado para os experimentos. É comum em uma organização que um número pequeno de recuarsos seja atribuído à cada papel, sendo estes suficientes para a simulação de um cenário de workflow.

A *adequação* é representada na tabela pela intersecção entre um recurso e uma tarefa (colunas **Tarefa A**, **Tarefa B**, **Tarefa C**) e indica o quão adequado um recurso é para executar uma determinada tarefa. Os valores de *adequação* podem estar em um intervalo $[0,1]$ e foram atribuídos aleatoriamente com valores do conjunto $\{0,4,\ 0,5,\ 0,6,\ 0,7,\ 0,8,\ 0,9,\ 1,0\}$.

Tabela 3: Mapeamento *papel* x *recurso* e adequação *recurso* x *tarefa*

		Adequação		
Papéis	**Recursos**	**Tarefa A**	**Tarefa B**	**Tarefa C**
papel1	r_1	1,0	0	0
	r_2	0,9	0	0
	r_3	0,8	0	0
	r_4	0,8	0	0
	r_5	0,8	0	0
papel2	r_6	0	0,8	0
	r_7	0	0,8	0
	r_8	0	0,6	0
	r_9	0	1,0	0
	r_{10}	0	0,7	0
papel3	r_{11}	0	0	0,4
	r_{12}	0	0	1,0
	r_{13}	0	0	0,9
	r_{14}	0	0	0,8
	r_{15}	0	0	0,5

5.2 Estratégias para a distribuição de tarefas

Sistemas de workflow realizam a distribuição de tarefas através dos mecanismos *Pull* e *Push* [13]. Estas estratégias apresentam resultados divergentes em relação a qualidade e tempo de execução do workflow. Enquanto a estratégia *Push* alcança altos níveis de qualidade na execução do workflow, a *Pull* obtém os menores intervalos de tempo de execução. Por exemplo, um sistema de workflow que adote o mecanismo *Push* de distribuição, devido a característica de entregar a tarefa somente ao recurso mais competente para executá-la, alcança altos índices de qualidade em prejuízo do tempo de execução, em contrapartida, o mecanismo *Pull* apresenta as tarefas ao conjunto dos recursos, e um destes recursos decide por executá-la, desta forma a distribuição é realizada em intervalos de tempo menores, porém com baixos níveis de qualidade.

A adoção de somente um dos dois mecanismos para a distribuição não é uma boa estratégia [11, 13, 15]. O sistema de workflow que adota o mecanismo *Push*, pode enviar uma tarefa para um recurso que não está disponível, assim a tarefa pode permanecer no sistema até que o determinado recurso esteja novamente disponível, acarretando em filas de espera. Por outro lado, um sistema que adote somente o mecanismo *Pull* pode ter as tarefas executadas por recursos que não são os mais aptos para tal, degradando assim a qualidade de execução das tarefas.

Outros mecanismos de distribuição foram concebidos no intuito de tentar criar um balanço entre qualidade e desempenho. Estes mecanismos foram definidos tendo como base as estratégias *Push* e *Pull*.

São considerados para o nosso experimento 12 estratégias de distribuição de tarefas. Entre as estratégias adotadas

estão inseridas as estratégias *Push* e *Pull*, as estratégias *Receptive, Typical* e *Selective* [11], as estratégias *Sel-push, Sel-push-10* e *Sel-push-ft* [15] e quatro novas estratégias propostas *Full-AbsAllocHITS, Random-AbsAllocHITS, Selective-AbsAllocHITS e Dynamic-AbsAllocHITS*.

Seja R o conjunto dos recusos, tal que, $R = \{r_1, r_2, ..., r_n\}$, e T o conjunto das tarefas, tal que, $T = \{t_1, t_2, ..., t_m\}$, os mecanismos definidos em conformidade com as estratégias *Push*, *Pull*, *Receptive*, *Typical*, *Selective*, *Sel-push*, *Sel-push-10* e *Sel-push-ft* realizam a distribuição da seguinte forma:

- *Push*: Um recurso r é escolhido pelo sistema para executar uma tarefa t;
- *Pull*: Uma tarefa t é visível a todos os recursos $r \in R' \subseteq R$ tal que R' é o subconjunto de recursos qualificados a executar t;
- *Receptive*: Uma tarefa t é visível somente aos recursos r que possuirem fator de alocação absoluta $fvab(t, r)$ acima de um limiar $0,5$. Este limiar indica que os recursos são receptivos;
- *Typical*: Uma tarefa t é visível somente aos recursos r que possuirem fator de alocação absoluta $fvab(t, r)$ acima de um limiar $0,4$. Este limiar indica que os recursos são típicos;
- *Selective*: Uma tarefa t é visível somente aos recursos r que possuirem fator de alocação absoluta $fvab(t, r)$ acima de um limiar $0,6$. Este limiar indica que os recursos são seletivos;
- *Sel-push*: Uma tarefa t é entregue ao recurso r mais apto, se r estiver disponível. Caso contrário, um recurso disponível com aptidão mais próxima de r é selecionado.
- *Sel-push-10*: Uma tarefa t é entregue ao recurso mais disponível entre o conjunto de recursos em uma vizinhança de 10% do valor $fvab(t, r)$, onde r é o recurso mais apto.
- *Sel-push-ft*: Uma tarefa t é entregue a um recurso r de forma semelhante à estratégia *Sel-push*, entretanto considera a *disponibilidade* e o tempo restante para concluir a tarefa atual.

5.3 Estratégias baseadas no algoritmo HITS

Quatro estratégias para distribuição de tarefas são propostas neste trabalho, todas tem como princípio a classificação dos recursos e tarefas através do algoritmo *AbsAllocHITS* que gera duas filas, a fila de recursos(*hubs*) e a fila de tarefas(*autoridades*). Essas filas estão ordenadas em ordem crescente dos valores de *hub* e *autoridade* e aqui são chamadas de *rankings*. Os primeiros elementos destes *rankings* são respectivamente os recursos mais especializados e as tarefas mais restritivas.

Consideramos que as tarefas mais restritivas e os recursos mais especializados trariam maior dificuldade para associação, então estas associações devem ser realizadas logo de início. À medida que os recursos vão sendo alocados, restam no *ranking* recursos mais versáteis, ou seja, recursos aptos a executar uma maior quantidade de tarefas distintas. Deste mesmo ponto de vista, a medida que as tarefas vão sendo executadas, restam tarefas menos restritivas, ou seja, tarefas que podem ser executadas por uma quantidade maior de recursos.

A distribuição de tarefas é realizada através de um algoritmo de distribuição. Este algoritmo obtém um conjunto de tarefas do sistema de *workflow*, associa os recursos do sistema às tarefas pertinentes e aplica a classificação por meio do algoritmo *AbsAllocHITS* apresentado na seção anterior. Após a classificação é selecionada uma tarefa e alocado um recurso para realização da tarefa. A seleção de tarefas e alocação de recursos prossegue iterativamente até que todas as tarefas sejam alocadas.

Variamos o algoritmo de montagem do grafo de entrada e a forma de seleção das tarefas para alocação, resultando em quatro estratégias que são descritas a seguir.

Full-AbsAllocHITS

A distribuição de tarefas através da estratégia *Full-AbsAllocHITS* é realizada conforme apresentado na figura 6 e descrita a seguir.

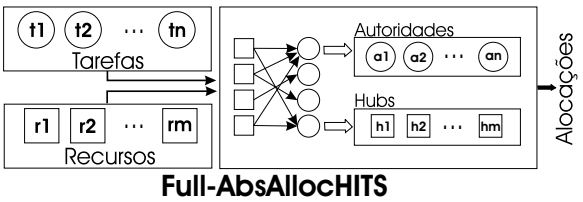


Figura 6: Estratégia *Full-AbsAllocHITS* de distribuição de tarefas.

Todas as tarefas coletadas pelo sistema e os recursos associados a pelo menos uma dessas tarefas formam o grafo de entrada para o algoritmo *AbsAllocHITS*. Este algoritmo gera então o *raking* de tarefas, em ordem crescente de seu grau de *hub* e o *ranking* de recursos, em ordem crescente de seu grau de *autoridade*.

A tarefa t_i, no topo do *ranking* é selecionada e removida do *ranking*. Então, o *ranking* de recursos é percorrido à partir do topo até que seja encontrado um recurso que esteja associados à tarefa t_i. Sendo r_j este recurso, a tarefa t_i é alocada ao recurso r_j. A distribuição prossegue iterativamente com a seleção de novo par (tarefa, recurso) até que todas as tarefas sejam alocadas.

Random-AbsAllocHITS
Nesta estratégia, o grafo de entrada para o algoritmo *AbsAllocHITS* é montado da mesma maneira que na estratéria *Full-AbsAllocHITS*. Entretanto, após a classificação, a seleção de uma tarefa ignora a ordem no *ranking* de tarefas e uma tarefa t_i é escolhida aleatoriamente. As demais etapas são iguais à estratégia *Full-AbsAllocHITS*.

Selective-AbsAllocHITS
Nesta estratégia, o grafo de entrada para o algoritmo *AbsAllocHITS* contêm apenas recursos com fator de alocação absoluta maior que o limiar de $0,6$ ($absAlloc(tarefa, recurso) > 0,6$) (veja figura 7). Recursos com essa característica são chamados *recursos seletivos* em [11]. As demais etapas são iguais à estratégia *Full-AbsAllocHITS*.

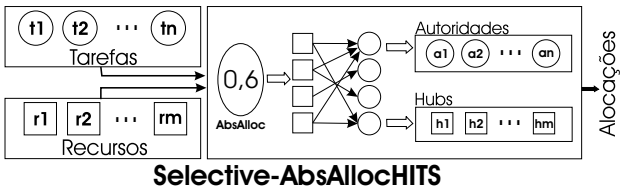


Figura 7: Estratégia *Selective-AbsAllocHITS* de distribuição de tarefas.

Dynamic-AbsAllocHITS
Nesta estratégia, procura-se inicialmente, montar o grafo de entrada para o algoritmo *AbsAllocHITS* com recursos com fator de alocação absoluta maior que o limiar de $0,6$ ($absAlloc(tarefa, recurso) > 0,6$). Entretanto, caso não exista recurso com fator de alocação absoluta que se enquadre neste limiar, o limiar é decrementado em $-0,1$, repetindo-se o decremento até que seja encontrado pelo menos um recurso com fator de associação à tarefa superior ao novo limiar(veja figura 8). As demais etapas são iguais à estratégia *Full-AbsAllocHITS*.

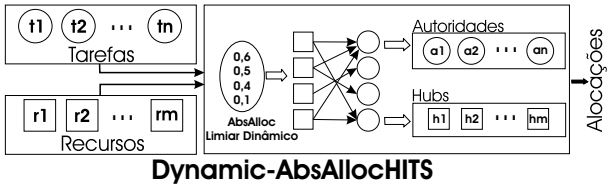


Figura 8: Estratégia *Dynamic-AbsAllocHITS* de distribuição de tarefas.

5.4 Ambiente de Experimentação

As estratégias de implementação do AbsAllocHITS foram simuladas pela ferramenta Lambari-SWTE (*Simple Workflow Test Environment*) [16]. O Lambari é um simulador de *workflow*, implementado em Java, que dá apoio à execução de vários processos simultâneos.

A figura 9 ilustra o funcionamento do simulador Lambari. Inicialmente, o Lambari analisa o arquivo XPDL[1] que contém dados relativos ao processo (fluxo de tarefas), aos recursos e às tarefas. O Lambari é capaz de simular todas as estratégias ou uma estratégia específica. Ao final da simulação, o Lambari apresenta o grafo de seqüenciamento de tarefas, a listagem correspondente a esse grafo, e gráficos comparativos entre as estratégias simuladas.

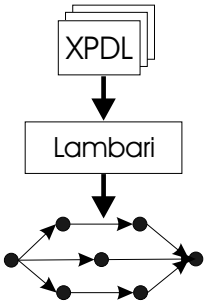


Figura 9: Funcionamento do Lambari-SWTE

A figura 10 mostra o grafo de seqüenciamento de tarefas oferecido pelo Lambari. O grafo em questão corresponde à simulação de quatro instâncias do processo da figura 5. Os vértices do grafo correspondem a tarefas. Os vértices **start** e **end** são tarefas fictícias que representam o início e o fim de todas as instâncias. Os arcos horizontais ligam tarefas de uma mesma instância. Assim, a seqüência de instâncias de tarefa *Start, A.1, B.1, C.1, End* corresponde à execução da instância 1 do processo citado. Os arcos verticais indicam que as tarefas ligadas por estas são executadas por um mesmo recurso (que não é apresentado explicitamente no grafo). Deste modo, as instâncias de tarefa *A.2 e A.4* são executadas pelo mesmo recurso, assim como as instâncias *B.1 e B.4*, e *C.3 e C.4*. Finalmente, os arcos estão rotulados com o tempo de execução da tarefa. A tarefa *Start*, por exemplo, demanda zero unidade de tempo, por sua vez, a tarefa *A.2* demanda 12 (doze) unidades de tempo.

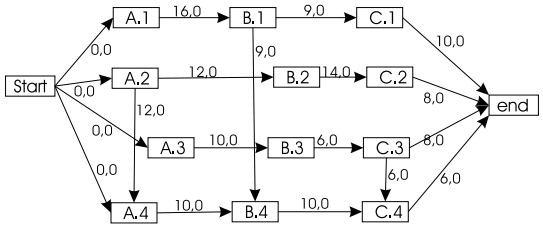


Figura 10: Grafo de sequenciamento.

6. RESULTADOS

As estratégias Full-AbsAllocHITS, Random-AbsAllocHITS, Selective-AbsAllocHITS e Dynamic-AbsAllocHITS são com-

[1]XML Process Definition Language [18]

paradas entre si e contra as estratégias *Push* e *Pull*, *Receptive* e *Selective* [11], e *Sel-push-10* [15].

Foram realizados experimentos para todas estratégias acima citadas. São apresentadas análises comparando as estratégias propostas neste trabalho com as da literatura. A primeira análise compara as estratégias baseadas no algoritmo *AbsAllocHITS* entre si. A segunda compara as estratégias *Push, Pull* e *Selective-AbsAllocHITS*. A terceira, compara as estratégias *Selective, Sel-Push-10* e *Selective-AbsAllocHITS*. Finalmente, é apresentada uma comparação global incluindo todas as estratégias acima.

Os gráficos foram traçados a partir da média obtida das 20 simulações realizadas em cada estratégia. Dois tipos de gráficos são apresentados: tempo de execução e qualidade. Os gráficos de tempo comparam carga de trabalho do *workflow* versus tempo de execução. Os gráficos de qualidade comparam carga de trabalho de *workflow* versus índice de qualidade. A carga de trabalho apresentada no gráfico corresponde àquela apresentada na tabela 2.

6.1 Análise das estratégias baseadas no algoritmo AbsAllocHITS

Os gráficos das figuras 11 e 12 apresentam a comparação entre as estratégias propostas neste trabalho. Percebe-se que a estratégia *Full-AbsAllocHITS* é a que demanda menor tempo de execução. Contudo, esta estratégia não apresenta um bom desempenho em termos de qualidade.

A estratégia que apresenta o melhor compromisso entre tempo e qualidade é a *Selective-AbsAllocHITS*. Assim sendo, esta estratégia foi selecionada como representante desta proposta para ser comparada com as estratégias representantes dos trabalhos relacionados.

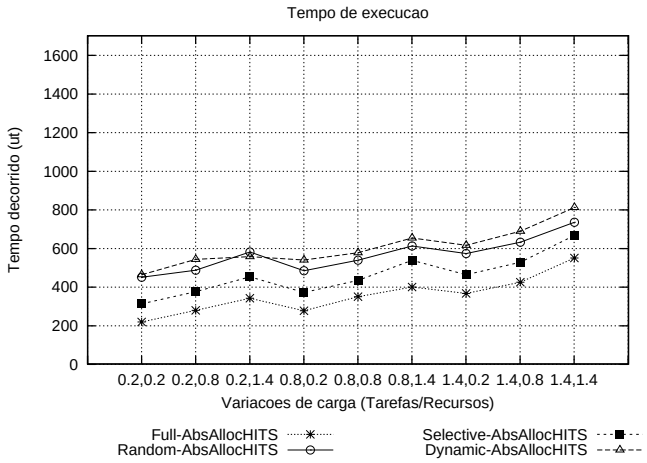


Figura 11: Gráfico "Tempo de execução" para as estratégias baseadas no algoritmo HITS.

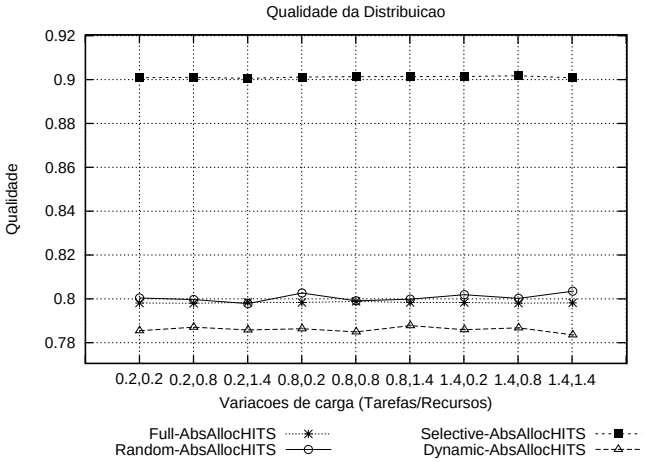


Figura 12: Gráfico "Qualidade da distribuição" para as estratégias baseadas no algoritmo HITS.

6.2 Comparação dos resultados das estratégias Pull, Push e Selective-AbsAllocHITS

As estratégias *Push* e *Pull* representam, respectivamente, limiares de tempo e qualidade. A literatura considera que a estratégia *Push* representa um limiar ideal de qualidade, enquanto a estratégia *Pull* como um limiar ideal de tempo. A comparação aqui apresentada tem o objetivo de situar a estratégia *Selective-AbsAllocHITS* entre esses limiares. Os gráficos das figuras 13 e 14 apresentam essa comparação.

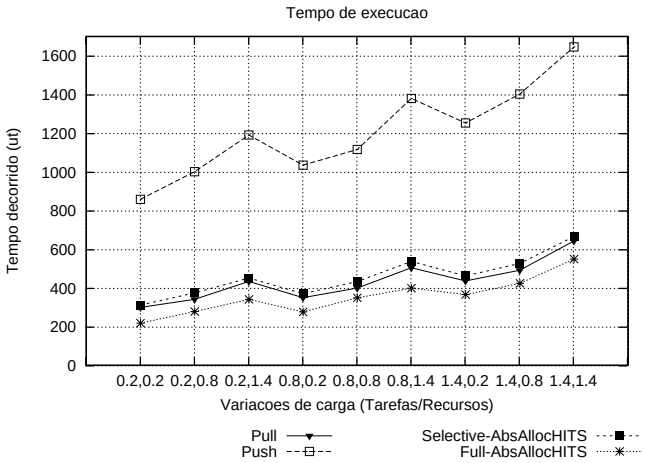


Figura 13: Gráfico "Tempo de execução" para as estratégias *Push, Pull e Selective-AbsAllocHITS*.

Observa-se que a estratégia *Selective-AbsAllocHITS* apresenta um desempenho acentuadamente superior em termos de tempo de execução quando comparada à estratégia *Push* e apresenta um desempenho próximo ao da estratégia *Pull*.

Em termos de qualidade pode-se perceber que a estratégia *Selective-AbsAllocHITS* se coloca entre os limiares estabelecidos pelas estratégias *Push* e *Pull*.

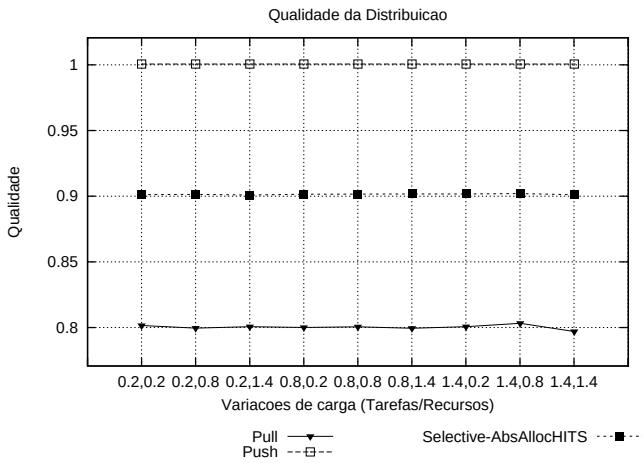


Figura 14: Gráfico "Qualidade da distribuição" para as estratégias *Push, Pull e Selective-AbsAllocHITS*.

6.3 Comparação dos resultados das estratégias Selective, Sel-Push e Selective-AbsAllocHITS

Uma outra comparação é realizada entre a *Selective-AbsAllocHITS* e as estratégias *Selective*, e *Sel-Push*, neste caso, o principal objetivo é de posicionar a estratégia *Selective-AbsAllocHITS* entre as melhores estratégias de distribuição apresentadas em trabalhos similares.

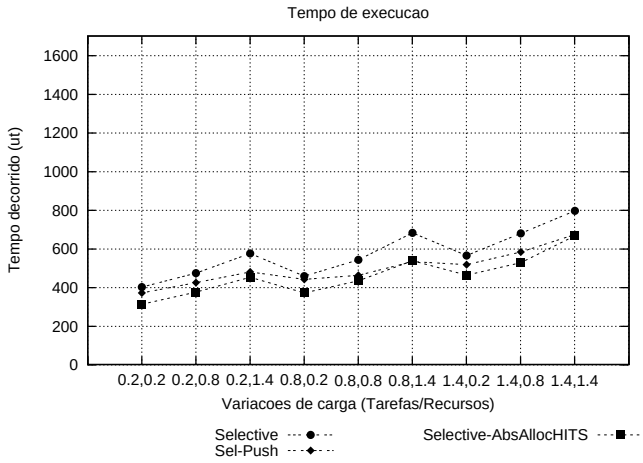


Figura 15: Gráfico "Tempo de execução" para as estratégias *Selective, Sel-Push e Selective-AbsAllocHITS*.

Os gráficos 15 e 16 comparam as estratégias *Selective*, *Sel-Push* e *Selective-AbsAllocHITS*. O gráfico da figura 15

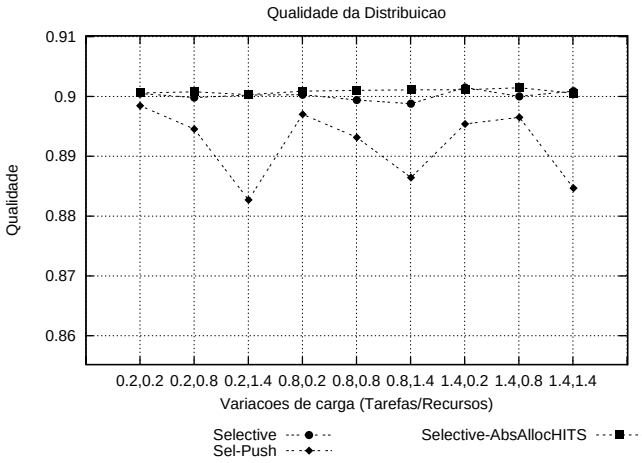


Figura 16: Gráfico "Qualidade da distribuição" para as estratégias *Selective, Sel-Push e Selective-AbsAllocHITS*.

mostra uma vantagem da estratégia *Selective-AbsAllocHITS* de aproximadamente 11% sobre a estratégia *Sel-Push*, no que se refere a tempo de execução. Por outro lado, a *Selective-AbsAllocHITS* apresenta desempenho superior relação à estratégia *Selective* de aproxiamadamente 25%.

A qualidade da execução da estratégia *Selective-AbsAllocHITS* alcançou patamares de aproximadamente 1.5% acima dos resultados dos mecanismos *Selective* e *Sel-Push*. Desta forma, pode-se concluir que a estratégia *Selective-AbsAllocHITS* mantém um maior compromisso entre desempenho e qualidade na execução de instâncias de *workflow* quando comparada a essas estratégias.

6.4 Análise global do tempo de processamento e da qualidade da distribuição

As figuras 17 e 18 demonstram os resultados obtidos por todas as simulações executadas.

O gráfico de tempo de execução apresentado na figura 17 mostra que a estratégia *Full-AbsAllocHITS* obteve o melhor desempenho neste quesito, sendo até melhor que a estratégia *Pull* citada como limite inferior em [11]. Ainda na figura 17 observa-se que as estratégias *Pull*, *Receptive* e *Selective-AbsAllocHITS*, nesta ordem, são as que apresentam menores índices de tempo para as simulações realizadas. Em contrapartida, o gráfico de qualidade da figura 18 indica que as estratégias *Push*, *Selective-AbsAllocHITS*, *Selective* e *Sel-Push-10*, nesta ordem, obtiveram os melhores patamares de qualidade do conjunto das estratégias testadas.

7. CONCLUSÃO

O presente trabalho associou os conceitos da análise de ligações (*Link Analysis*) a conceitos de distribuição de tarefas em sistemas de workflow. Pode-se destacar que a es-

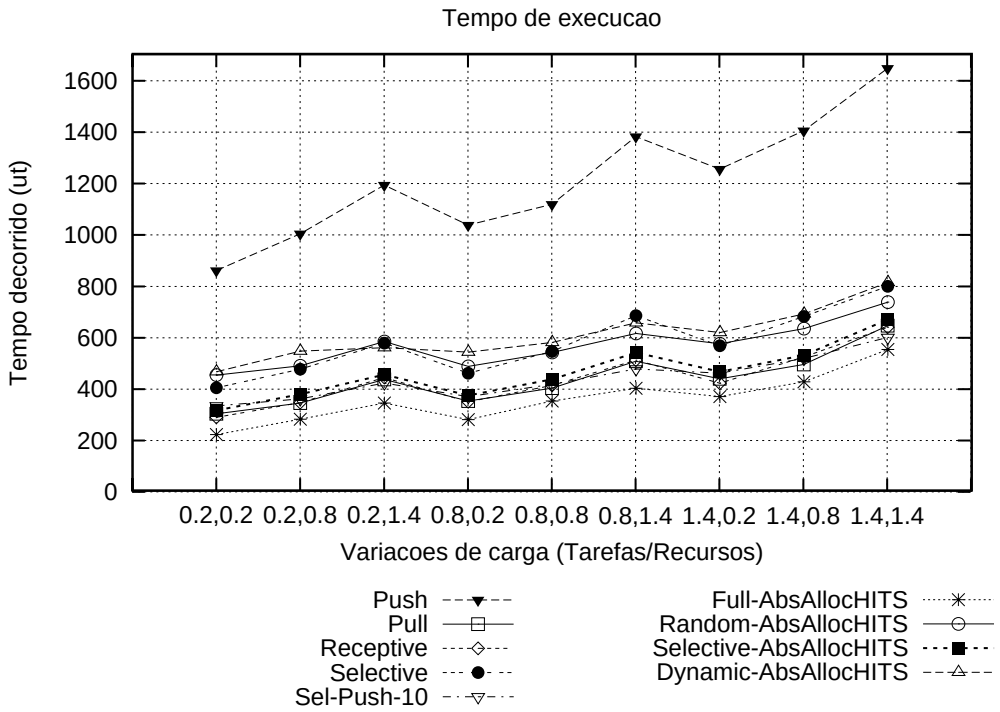


Figura 17: Gráfico "Tempo de execução".

tratégia *Full-AbsAllocHITS* obteve tempos de execução inferiores entre aquelas comparadas neste trabalho.

Além disso, o mecanismo *Selective-AbsAllocHITS* alcançou o melhor balanceamento entre qualidade e tempo de execução, quando comparado com os trabalhos relacionados aqui apresentados. Os resultados obtidos com os experimentos mostraram que esta abordagem obteve ganhos de desempenho de 11% em relação aos resultados apresentados por Veloso e Macêdo, e de 25% em relação aos resultados de Kummar et al.

8. AGRADECIMENTOS

Os dois primeiros autores agradecem à FAPEMIG (Fundação de Apoio à Pesquisa do Estado de Minas Gerais) o apoio financeiro parcial a este trabalho no contexto dos projetos EDT 312/05 e EDT 322/05.

9. REFERÊNCIAS

[1] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Trans. Inter. Tech.*, 5(1):231–297, 2005.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. of the 7th International World Wide Web Conference (WWW7)*, pages 107–117, Brisbane, Australia, 1998.

[3] P. Calado, B. Ribeiro-Neto, N. Ziviani, E. Moura, and I. Silva. Local versus global link information in the web. *ACM Transactions on Information Systems*, 21(1):42–63, january 2003.

[4] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the web's link structure. *Computer*, 32(8):60–67, 1999.

[5] C. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. Pagerank, hits and a unified framework for link analysis. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 353–354, New York, NY, USA, 2002. ACM Press.

[6] O. Drori. Algorithm for documents ranking: idea and simulation results. In *SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 99–102, New York, NY, USA, 2002. ACM Press.

[7] M. Henzinger. Hyperlink analysis on the world wide web. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 1–3, New York, NY, USA, 2005. ACM Press.

[8] M. R. Henzinger. Hyperlink analysis for the web. *IEEE Internet Computing*, 5(1):45–50, 2001.

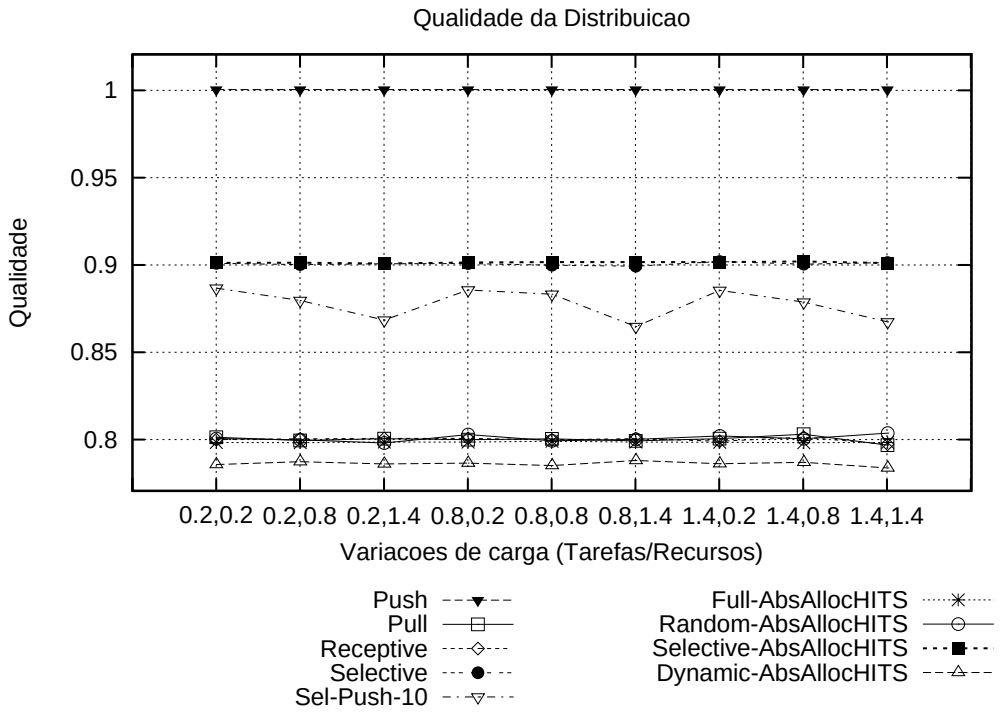


Figura 18: Gráfico "Qualidade da distribuição".

[9] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM-SIAM (Symposium on Discrete Algorithms)*, 46(5):604–632, 1999.

[11] A. Kumar, W. M. van der Aalst, and E. M. Verbeek. Dynamic work distribution in workflow management systems: How to balance quality and performance? *Journal of Management Information Systems*, 18(3):157–193, 2001.

[12] I. Silva, B. Ribeiro-Neto, P. Calado, E. Moura, and N. Ziviani. Link-based and content-based evidential information in a belief network model. In *Proc of the 23rd ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000. Best student paper.

[13] G. B. Tramontina. Análise de problemas de escalonamento de processos em workflow. Master's thesis, Instituto de Computação, Universidade de Campinas (UNICAMP), Campinas, SP, Brazil, 2004.

[14] W. M. P. van der Aalst and K. M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT Press, 2002.

[15] R. R. Veloso. Distribuição de tarefas em sistemas de workflow baseada na aptidão dos recursos. Master's thesis, Faculdade de Computação, Universidade Federal de Uberlândia (UFU), Uberlândia, MG, Brazil, 2006.

[16] R. R. Veloso and A. Macedo. Lambari: Simple workflow test environment. Technical report, Faculdade de Computação, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2005.

[17] K. Wang and M.-Y. T. Su. Item selection by "hub-authority" profit ranking. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 652–657, New York, NY, USA, 2002. ACM Press.

[18] WfMC. Terminology and glossary. Technical Report WfMC-TC-1011, Workflow Management Coalition, Hampshire, Reino Unido, Fevereiro 1999.

[19] W. Xing and A. Ghorbani. Weighted pagerank algorithm. In *CNSR '04: Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04)*, pages 305–314, Washington, DC, USA, 2004. IEEE Computer Society.

ThreadMap: Uma Abordagem para Mapeamento de Mensagens de Correio Eletrônico

Hélio Oliveira Queiroz Junior
Secretaria da Fazenda do Estado da Bahia - SEFAZ
Avenida 2, 260
Salvador, Ba 41745-003 Brasil
helio.queiroz@gmail.com

Manoel Gomes Mendonça Neto
Universidade Salvador – UNIFACS
Rua Ponciano de Oliveira, 126
Salvador, Ba 41950-275 Brasil
mgmn@unifacs.br

RESUMO

O correio de eletrônico é a ferramenta de colaboração mais difundida dentro das organizações, sendo utilizado não apenas para a comunicação de informativos e recados, mas também para desenvolver discussões não-presenciais e assíncronas entre membros de uma equipe. Define-se um *thread* como um conjunto de mensagens trocadas entre indivíduos, tipicamente em torno de um mesmo assunto, em uma ordem cronológica, através de respostas e encaminhamentos, simulando uma conversa presencial. Através do mapeamento de *threads* é possível identificar características tais como a duração típica dessas conversas, a quantidade de mensagens trocadas e a quantidade de participantes envolvidos. Este artigo apresenta uma solução para o mapeamento de *threads* baseado na interceptação direta de mensagens no servidor de correio eletrônico e nos dados contidos no cabeçalho das mensagens. Além da solução, este artigo apresenta uma implementação utilizando um servidor de mercado, uma interface que permite a visualização dos *threads* mapeados juntamente com as respectivas mensagens trocadas, e um estudo de caso realizado na Gerência de Tecnologia da Secretaria da Fazenda do Estado da Bahia, que demonstrou a viabilidade da solução proposta.

Palavras-chave

Thread, Correio Eletrônico, Mensagem Eletrônica, Visualização, Mapeamento.

ABSTRACT

The e-mail is widespread in modern organizations. They are used not only for sending information, but also to develop full fledge non-presential and asynchronous discussions. These discussions are commonly referred as threads. Formally, a Thread is defined as a set of electronic message exchanged between individuals, typically around a given subject, in chronological order, through replies and forwards, simulating a presential conversation. Through thread mapping, it is possible to analyze features like typical duration of threads, number of exchanged messages, and number of participants involved in a thread. This paper proposes a solution for thread mapping based on intercepting messages directly at the e-mail server and on the analysis of the header of these messages. Beyond the proposal, this article presents an implementation of the approach for a well-know e-mail server, an interface for visualizing the mapped threads and their messages, and a case study on the usage of the implemented approach. The case study was run in a real organization, the Technology Sector of the State of Bahia Revenue Services. The study showed that the proposed solution was viable and useful.

Keywords

Thread, Electronic Mail, Electronic Message, Visualization, Mapping.

1. INTRODUÇÃO

Atualmente, o correio eletrônico é uma das ferramentas de trabalho mais utilizadas dentro das organizações. Muitos dos processos organizacionais hoje são totalmente dependentes da troca de mensagens via correio eletrônico. Desde solicitações de informações e autorizações até decisões importantes estão registradas nas trocas de mensagens.

Nesse contexto, milhares de mensagens eletrônicas são trocadas entre indivíduos, permitindo que estes estejam participando, de uma forma simultânea e assíncrona, de diversas conversas ou discussões, entre grupos distintos de pessoas.

As mensagens trocadas de forma eletrônica simulam uma conversa presencial, com a vantagem de que toda ela pode ser armazenada de forma persistente, servindo como uma base de conhecimento explícito ou meramente um registro da execução de um processo organizacional.

Analisando do ponto de vista da organização, o mapeamento e visualização dessa troca de mensagens podem ser utilizados para identificar processos ainda não formalizados, para verificar como se dá o fluxo de mensagens dentro de um grupo de trabalho, auxiliar o mapeamento de redes sociais e analisar a sua interação.

Para os usuários de correio eletrônico, a possibilidade de visualizar de forma rápida, integrada e consistente uma trilha de mensagens minimiza bastante o tempo necessário para o usuário conseguir se posicionar e interagir dentro da conversa em andamento, principalmente em função do grande volume de mensagens de correio eletrônico trocadas diariamente.

SBSC Simpósio Brasileiro de Sistemas Colaborativos

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 118-128.

Este artigo está organizado em nove seções. Após esta introdução, serão apresentadas três seções sobre *threads* de mensagens e revisão bibliográfica das técnicas de mapeamento e visualização de *threads*. As próximas três seções tratam da solução *ThreadMap*, de sua implementação utilizando um servidor de correio eletrônico de mercado e do estudo de caso realizado. Na seqüência, serão vistas a conclusão do trabalho e as referências utilizadas.

2. *THREADS* DE MENSAGENS

A troca de mensagens, utilizando o correio eletrônico, permite aos seus usuários a participação em diversas discussões sobre temas variados, de forma simultânea. Essa troca de mensagens entre usuários, normalmente sobre o mesmo assunto é chamada de *Thread*.

Lewis e Knowles [4] definem um *thread* como uma conversação entre duas ou mais pessoas através da troca de mensagens. Para Palme [8], *threads* são conjuntos de mensagens que são, direta ou indiretamente, respostas a outras mensagens.

De acordo com o conceito de Palme [8], uma vez que as mensagens de um *thread* são sempre fruto de respostas a outras mensagens, um *thread* pode ser visto como uma árvore, onde a primeira mensagem do thread é a raiz da árvore, conforme mostrado na Figura 1.

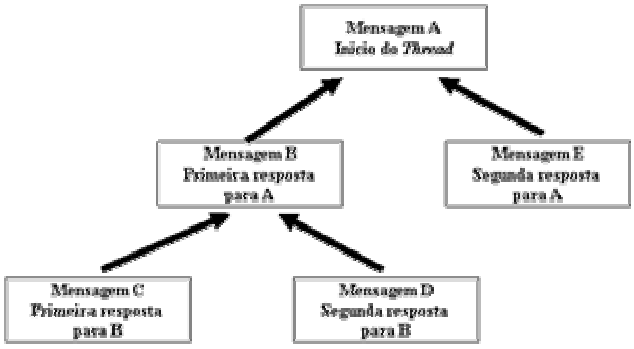


Figure 1. Exemplo de *Thread* – Adaptado de Palme[8]

Segundo Nenkova e Bagga [6], os *threads* de correio eletrônico são a forma mais comum de representação e armazenamento de grupos de discussão ou listas de distribuição, principalmente em função do grande volume de mensagens trocadas nestes serviços.

As definições apresentadas por Lewis e Knowles [4] e Palme [8] sobre a simulação de uma conversa presencial via correio eletrônico, juntamente com a consideração de Nenkova e Bagga [6] sobre a representação para grupos de discussão e listas de distribuição apontam para possibilidade de registro, mapeamento, e posterior visualização e análise desses *threads*.

3. MAPEAMENTO DE *THREADS*

O processo de mapeamento de *threads* de mensagens eletrônicas pode ser realizado através de diferentes algoritmos, a depender dos objetivos desse mapeamento e da fonte de informações, no nosso caso, as mensagens eletrônicas trocadas entre usuários, disponíveis para esse fim.

Quando a técnica de mapeamento de *thread* utiliza a caixa postal de um usuário como fonte de informações, os objetivos estão focados nas necessidades do usuário, principalmente na interação com o grande volume de mensagens na sua caixa postal. Neste contexto, através da visualização de *threads*, torna-se mais fácil localizar a seqüência de mensagens trocadas na conversação, permitindo ao usuário se posicionar na conversa de forma mais rápida.

Nesse caso, objetivos organizacionais como identificação de processos não formais e análise do fluxo de informações dentro do grupo de trabalho são prejudicadas, pois não existe um mapeamento global de mensagens. Não há como garantir que o usuário tenha participado de todo o *thread* e que ele não tenha excluído nenhuma mensagem da conversação.

Uma alternativa é a utilização de técnicas de mapeamento baseadas no repositório de mensagens trocadas em listas de discussões. Os algoritmos são basicamente os mesmos utilizados no processo de mapeamento usando a caixa postal de um usuário como fonte. Nesse caso, tem-se a garantia de que todo o *thread* é mapeado, mas normalmente esses *threads* são públicos ou muito abrangentes, dificultando o alcance de objetivos organizacionais específicos. Outro aspecto importante é que essa fonte é normalmente associada a um tema, o que dificulta o mapeamento de troca de informações dentro de um grupo de trabalho.

Outra alternativa é a interceptação do evento de chegada de mensagens em caixas postais no servidor de correio eletrônico da organização. Essa técnica não tem as limitações de abrangência da caixa postal de um usuário, nem tão pouco a especificidade de uma lista de discussão. Nessa técnica, o *thread* é mapeado a cada troca de mensagem dentro de um servidor, facilitando o alcance dos objetivos organizacionais já discutidos. Como desvantagem, essa técnica é dependente da existência de um mecanismo de interceptação de eventos no servidor de correio eletrônico utilizado.

A Tabela 1 apresenta um resumo sobre as vantagens e desvantagens de cada uma das alternativas para o mapeamento de *threads*.

Tabela 1. Vantagens e desvantagens das abordagens de mapeamento de *threads*

Abordagem de Mapeamento	Vantagens	Desvantagens
Caixa postal de usuário	* Foco na necessidade do usuário; * Auxílio na interação com o grande volume de mensagens; * Facilita a localização do usuário na conversação em andamento.	* Dificuldade na identificação de processos não formais; * Mapeamento do *thread* com a visão particular do usuário; * Grande possibilidade de perda de trechos do *thread*;
Lista de	* Garantia de que todo	* Threads são

Abordagem de Mapeamento	Vantagens	Desvantagens
discussão	o *thread* é mapeado; * Mapeamento *thread* com visão global;	públicos ou muito abrangentes; * Fonte normalmente associada a um tema específico.
Interceptação de mensagem	* Garantia de que todo o *thread* é mapeado; * Mapeamento *thread* com visão global; * Não possui o especificidade das listas de discussões; * Auxilia na identificação de processos não formais.	* Dependência de mecanismo de interceptação de mensagens no servidor de correio eletrônico.

Seja qual for a origem da informação, as técnicas de mapeamento de *thread* utilizam informações contidas na mensagem eletrônica. Essas informações podem estar localizadas no corpo ou no cabeçalho da mensagem.

Utilizar as informações contidas no corpo da mensagem permite a identificação e agrupamento de acordo com o assunto tratado na mensagem. Como normalmente as ferramentas cliente de correio eletrônico mantêm o texto original ao enviar uma mensagem resposta, a análise do corpo da mensagem pode indicar a que *thread* ela pertence. Lewis e Knowles [4] apresentam um estudo onde o *thread* é mapeado utilizando a similaridade de vocabulário existente entre os textos das mensagens trocadas, com uma eficácia de 71% na identificação das mensagens pai.

Alguns autores apontam técnicas de mapeamento baseadas no cabeçalho das mensagens. Por exemplo, Palme [8] apresenta como alternativa para o mapeamento de *threads* o uso de informações contidas no cabeçalho das mensagens eletrônicas (*headers*). Nesta técnica, são utilizados os cabeçalhos *Message-Id, In-Reply-To e References* para realizar o mapeamento do *thread*.

Zawinski [14] apresenta um algoritmo que também faz uso dos dados contidos no cabeçalho das mensagens eletrônicas para o mapeamento de *threads*. Ele utiliza os campos *In-Reply-To, References e Subject* para agrupar as mensagens do mesmo *thread*. No caso da manipulação do assunto da mensagem, é necessária a supressão de prefixos normalmente incluídos pelas ferramentas clientes de correio eletrônico, nas operações de resposta e encaminhamento de mensagens. Esses prefixos são: *"Re:", "Res:", "Fw:", "Fwd:"*, dentre outros.

IMAP Extension Work Group [2] trata de uma proposta elaborada pelo IETF (*Internet Engineering Task Force*) para extensão do protocolo IMAP para ordenação e agrupamento de *thread*, na execução de buscas de mensagens disponíveis no servidor de correio eletrônico.

Essas extensões complementam o comando SEARCH disponível no protocolo IMAP. Com relação ao agrupamento dessas mensagens em threads, essa extensão suporta a implementação de dois algoritmos, definidos pelas palavras reservadas ORDEREDSUBJECT e REFERENCES.

No caso do algoritmo ORDEREDSUBJECT, as mensagens pesquisadas são ordenadas pelo assunto base (campo assunto com a remoção de palavras não significativas) e então pela data de envio. As mensagens são então agrupadas em *threads*, onde cada *thread* possui o mesmo assunto base.

As mensagens do *thread* são então ordenadas por data e a primeira mensagem trocada é considerada a mensagem raiz do *thread*. As demais mensagens são todas consideradas filhas da mensagem raiz, não existindo então mais de um nível no *thread* quando o algoritmo ORDEREDSUBJECT é utilizado. Percebe-se que nessa implementação muitas informações sobre as trocas de mensagens são perdidas.

O algoritmo REFERENCES é baseado na proposta de Zawinski [14]. Esse algoritmo agrupa as mensagens, estabelecendo entre elas um relacionamento pai/filho baseado nas respostas e trocas de mensagens realizadas. Esse relacionamento pai/filho é construído usando dois métodos: reconstruindo a árvore ancestral da mensagem utilizando o campo *references* do seu cabeçalho ou tomando como base o assunto (*subject*) das mensagens trocadas.

Nas diversas alternativas apresentadas, a solução possui como requisito a disponibilidade de um repositório com todas as mensagens a serem mapeadas, o que nem sempre é possível obter, uma vez que normalmente, vários usuários participam de uma conversação através do correio eletrônico. Além disso, a fonte de dados é um retrato (*snapshot*) da caixa postal de um usuário e reflete a sua participação, ou seja, a visão que um indivíduo tem sobre a conversação realmente realizada.

Fisher e Moody [1] apresentam um estudo envolvendo a captura de mensagens trocadas por 74 usuários durante 60 dias, utilizando como fonte para o mapeamento dos *threads* uma cópia da caixa postal de cada usuário e evidências como data e assunto para associação dessas mensagens aos *threads*. Além de dados estatísticos como número médio de mensagens trocadas nos *threads*, número de usuários envolvidos e tempo médio de resposta às mensagens trocadas, a identificação de que cada usuário possui uma percepção diferente sobre o mesmo *thread* merece destaque.

Pela técnica aplicada, além das diferentes visões, alguns trechos do *thread* podem ter sido perdidos, pois algumas mensagens já podem ter sido excluídas das caixas postais dos envolvidos. Tudo isso dificulta muito a possibilidade de uso dessas informações coletadas para a identificação de processos não formais e análise do fluxo de informações dentro de um grupo de trabalho.

As técnicas de mapeamento vistas apresentam em comum a característica de possuírem como fonte de dados um repositório com as mensagens já trocadas, ou seja, a montagem do *thread* é realizada após a conversação. Como conseqüência da fonte utilizada, as técnicas apresentam uma visão particular da conversação, com possibilidade de perda de trechos do *thread*.

Para minimizar os problemas encontrados, a alternativa de mapeamento de *threads*, baseado na interceptação das mensagens eletrônicas recebidas pelo servidor de correio eletrônico permitiria o mapeamento de todo o *thread*, independente de como essas mensagens estão armazenadas e organizadas em cada caixa postal de usuário. Essa alternativa,

entretanto, tem como requisito a captura das mensagens no momento em que elas são trocadas, não sendo possível a sua recuperação a posteriori.

Nesse caminho de solução surgem dois desafios: a identificação do *thread*, ou seja, a associação correta de cada mensagem recebida ao seu respectivo *thread* e qual a posição dessa mensagem dentro da cadeia de mensagens trocadas, uma vez que cada mensagem está frequentemente associada (sendo uma resposta ou encaminhamento) a uma mensagem pai dentro de um processo assíncrono.

4. VISUALIZAÇÃO DE *THREADS*

Existem diversas propostas de visualização de *threads* de mensagens. Em sua grande maioria, são apresentadas como uma evolução das ferramentas cliente de correio eletrônico, com o objetivo de facilitar a participação do usuário na conversação, assim como também melhorar a visualização global do *thread* e sua distribuição no tempo.

No caso das listas de discussão, é comum a apresentação de *threads* de mensagens sob o formado de árvore, onde cada mensagem de resposta é organizada de forma endentada e as mensagens são representadas pelo remetente e horário de postagem, sendo o título um apontador para o corpo da mensagem propriamente dito.

Uma variação dessa estrutura tradicional de árvore foi apresentada por Newman [7], chamada de *Narrow Tree*, cuja representação limita o processo de endentação das mensagens do *thread* a apenas um nível. Nessa técnica, as mensagens do *thread* são separadas por linhas divisórias de acordo com a sua posição dentro do *thread*. Esta técnica permite uma visão geral do *thread* e inclui o trecho inicial da mensagem na sua visualização.

Newman [7] apresenta uma outra técnica de visualização de *threads*, ainda baseada na árvore tradicional, mas em uma estrutura onde as mensagens eletrônicas são organizadas em linhas e colunas, como em uma tabela chamada *TreeTable*. Nela, cada célula representa uma mensagem do *thread* e as mensagens que são respostas diretas a uma mensagem pai estão organizadas em colunas irmãs na linha subseqüente. Todas as respostas são agrupadas ocupando a mesma área horizontal da mensagem pai.

Rohall e Gruen [11] apresentam o *ReMail: A reinvented email prototype*, que incorpora novas maneiras de visualizar, gerenciar e interagir com *threads* e outros grupos de mensagens relacionadas.

A proposta de Rohall e Gruen [11] suporta a visualização de *threads* em árvores, em paralelo com a mensagem. Quando uma mensagem é selecionada, as demais mensagens do *thread* são mostradas em um tom diferente e o nó correspondente a mensagem é selecionado na árvore do *thread*, localizada lado esquerdo da janela de trabalho do usuário.

As formas de visualização apresentadas até o momento possuem como característica a ênfase no relacionamento pai-filho entre as mensagens, mas essas técnicas não facilitam a determinação das relações temporais entre as mensagens trocadas.

Para permitir uma melhor visualização das relações de tempo entre as mensagens trocadas, Rohall [10] apresenta uma técnica chamada *TimeLine*, onde as linhas verticais representam fronteiras temporais, no caso, de um dia. O *thread* é visualizado como uma árvore disposta na horizontal, onde a dimensão tempo não é necessariamente linear, ou seja, dias com pouca ou nenhuma atividade podem ser mostrados de forma comprimida, permitindo que uma maior faixa de tempo esteja visível na tela do computador.

Venólia e Neustaedter [12] apresentam um modelo misto para visualização de conversação baseada em mensagens eletrônicas. Esse modelo, segundo os autores, foi desenhado para permitir que grandes *threads* de mensagens possam estar disponíveis e facilmente visíveis para o usuário, com pouca necessidade de rolamento vertical da tela visual. Essa técnica usa o princípio de visão geral mais detalhes, onde em uma única janela temos dois *frames*, um com os resumos e outro com os detalhes do item selecionado no *frame* resumo.

Kerr [3] propôs uma técnica de visualização chamada *Thread Arcs*, onde cada mensagem do *thread* é representada por um nó e os relacionamentos (respostas a mensagens) representados por arcos interligando os nós.

Nessa técnica, as mensagens são colocadas em seqüência, da esquerda para direita, de acordo com a sua ordem cronológica. Em seguida, os relacionamentos de respostas a mensagens são adicionados interligando os nós (mensagens) utilizando arcos.

5. A SOLUÇÃO *THREADMAP*

O *ThreadMap* é uma proposta de solução para o problema do mapeamento de *threads* de mensagens eletrônicas, focada na interceptação de mensagens no servidor de correio eletrônico como forma de minimizar os problemas anteriormente identificados no mapeamento de *threads* a partir da caixa postal de um usuário. Seu principal benefício é a capacidade de mapear toda a conversação, independente da visão de cada usuário, e minimizando o risco de perda de trechos do *thread*.

5.1 Funcionamento e Algoritmo

Uma solução de interceptação de mensagens possui dois desafios principais: a identificação do *thread*, ou seja, a associação correta de cada mensagem recebida ao seu respectivo *thread*; e a identificação da mensagem pai, ou seja, o posicionamento da mensagem dentro da cadeia de mensagens trocadas nesse *thread*, uma vez que toda mensagem está associada a uma mensagem pai, seja na forma de resposta ou encaminhamento.

Para resolver os problemas acima, essa solução propõe a inclusão de uma marca na mensagem de forma a identificar o *thread* da mensagem atual e a sua posição dentro da cadeia de mensagens da conversação.

Com essa marca, é possível determinar a qual *thread* a mensagem pertence. Além da identificação do *thread*, a marca registra qual a mensagem anterior na cadeia de troca de mensagens, para a correta associação do relacionamento pai e filho em cada *thread*. A ausência de marca indica o início de um novo *thread*. A Figura 2 apresenta o algoritmo que implanta a solução proposta.

De acordo com esse algoritmo, para cada mensagem capturada é verificada a que *thread* ela pertence e qual a mensagem pai, ou seja, a mensagem origem a partir da qual um determinado usuário enviou uma resposta ou um encaminhamento desta. As informações sobre o *thread* registram o usuário que o iniciou, a sua duração (data e hora de início e fim), os usuários envolvidos, a quantidade de mensagens trocadas e o intervalo entre elas, o assunto da conversação e quais foram as mensagens trocadas.

Para cada mensagem capturada

Verifica-se se a mensagem capturada já possui a marca do Thread

Se não possui a marca (inicio de um novo thread)

Registra o inicio do thread;
Armazena as informações sobre o thread e a mensagem capturada;
Insere a marca na mensagem capturada, com o número do novo thread e da mensagem atual, que será a mensagem pai das respostas e encaminhamentos;

Caso contrário (nova mensagem de thread já iniciado)
Obtém a informação do thread atual e da mensagem pai a partir da marca da mensagem;
Armazena as informações sobre a mensagem capturada;
Atualiza a marca da mensagem com o novo identificador da mensagem.
Fim

Figure 2. Algoritmo da solução *TheadMap*

As informações sobre cada mensagem capturada incluem o usuário emissor, os usuários receptores (diretos e indiretos), o assunto, a data e hora do envio, a que *thread* ela pertence, qual a sua mensagem pai (origem) e se houve ou não arquivos anexados. Para preservar a confidencialidade das informações contidas nas mensagens, essa solução não armazena o conteúdo da mensagem, sendo todas as informações acima relacionadas obtidas a partir do cabeçalho da mensagem.

A Tabela 2 mostra as informações colhidas de cada mensagem e as respectivas fontes nos campos do cabeçalho.

Esta solução se baseia na captura de mensagens e mapeamento do *thread* em tempo real e tem como requisito para a sua implementação que o servidor de correio eletrônico possua um mecanismo de suporte a eventos que permita a interceptação, captura e manipulação das mensagens recebidas.

Tabela 2. Fonte das informações colhidas sobre as mensagens interceptadas

Informação	Fonte
Identificador da mensagem	*Message-ID (cabeçalho)*
Identificador do *thread*	Identificador do thread contido na marca criada pelo algoritmo.
Identificador da mensagem pai (origem)	Identificador da mensagem pai (origem) na marca criada pelo algoritmo
Assunto	*Subject (cabeçalho)*
Data e hora da mensagem	*Date (cabeçalho)*
Usuário emissor	*From (cabeçalho)*
Usuários receptores	*To e Cc (cabeçalho)*
Existência de arquivos anexados	*Attachment (cabeçalho)*

5.2 Benefícios da solução

Como descrito anteriormente, essa solução apresenta uma proposta de mapeamento de *threads* baseado na interceptação de mensagens no servidor de correio eletrônico, minimizando a possibilidade de perdas de trechos do *thread* em função de uma possível distribuição geográfica desses dados nas diversas caixas postais dos usuários envolvidos.

Com base na solução proposta, é possível a realização do mapeamento completo de todos os *threads*, cujas mensagens sejam trocadas no âmbito de um servidor de correio eletrônico. Esta solução pode ser ampliada para permitir o mapeamento além das fronteiras de um servidor, desde que instalada em todos os servidores de correio eletrônico dos usuários envolvidos e utilizando uma base de dados centralizada. Para tal, a solução proposta pelo *ThreadMap* pode ser implementada para qualquer servidor de correio eletrônico para permita a interceptação da mensagens e o acesso aos seus dados.

A partir das informações de cada *thread*, estão disponíveis dados que permitem mapear a estrutura em árvore de mensagens trocadas entre os envolvidos.

Além dos dados sobre cada mensagem, podem ser obtidas informações como a duração de cada *thread*, a quantidade de participantes (usuários) envolvidos, a quantidade de mensagens trocadas de cada *thread*, e realizar análises sobre a distribuição estatística desses dados dentro do universo amostrado.

Outro aspecto que pode ser explorado diz respeito à participação dos indivíduos. Sob essa ótica, é possível mapear a participação de cada usuário na conversação e o fluxo da comunicação dentro dos grupos de trabalho, sendo possível identificar os usuários que mais contribuem nessas comunicações, a intensidade da comunicação entre dois indivíduos e o tempo médio de resposta desses usuários no momento de sua contribuição [13].

6. IMPLEMENTAÇÃO *THREADMAP*

Como já dito anteriormente, a solução *ThreadMap* pode ser implementada em qualquer servidor de correio eletrônico que possua um mecanismo de suporte a eventos, permitindo a interceptação, captura e manipulação das mensagens recebidas.

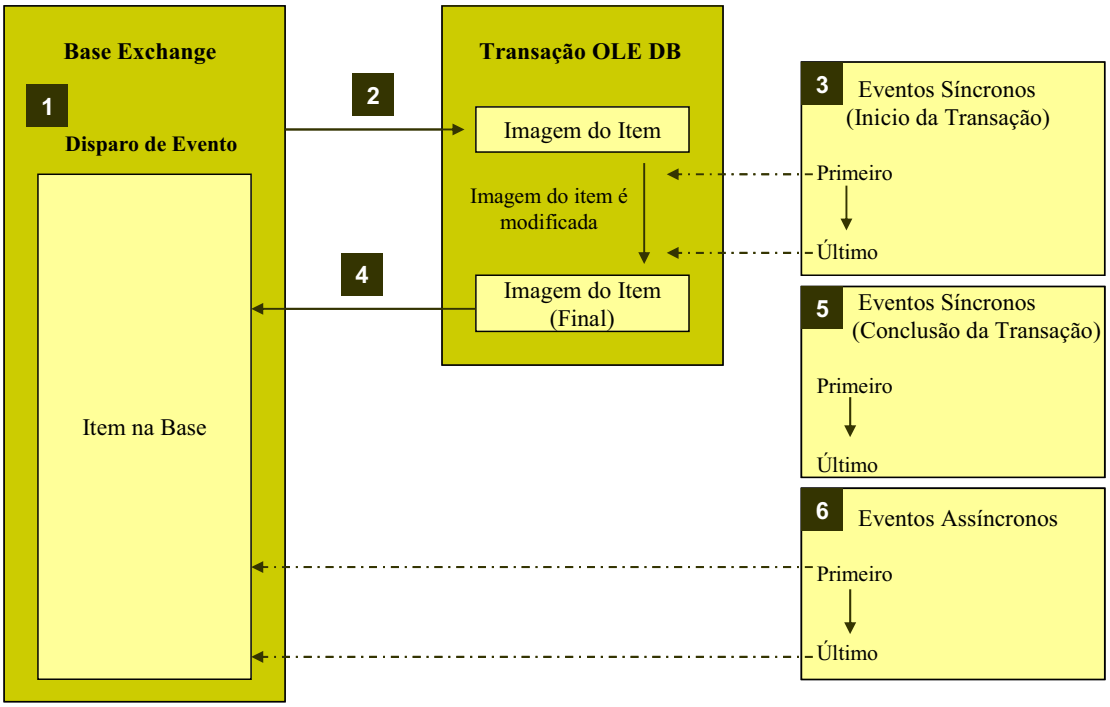


Figure 3. Diagrama de disparo dos eventos do *MS Exchange Server* [5]

Esta seção apresenta uma implementação *ThreadMap*, utilizando o servidor *MS Exchange Server* e utiliza o *Exchange Store Events* como mecanismo para interceptação das mensagens trocadas.

6.1 *Exchange Store Events*

O *MS Exchange Server* [5] possui um mecanismo chamado *Exchange Store Event*, que permite a programação de regras de negócio associadas a eventos, como por exemplo: salvar, excluir, modificar, copiar ou mover itens na base de dados do *MS Exchange* Server.

O *Exchange Store Event* suporta 3 (três) tipos de eventos: síncronos, assíncronos e de sistema. Os eventos síncronos e assíncronos são relativos a operações como o armazenamento ou remoção de itens na base de dados. Os eventos síncronos são disparados antes da conclusão (*commit*) da operação, enquanto que os eventos assíncronos são disparados após a conclusão das operações de armazenamento ou remoção dos dados. Os eventos de sistema são disparados em horários programados, ou nos eventos de partida e encerramento da base de dados do *MS Exchange Server*.

A partir dos eventos disponíveis, é possível a criação de código aplicativo, sob o formato de *Dynamic Link Library* (DLL) que são disparados na ocorrência desses eventos e são chamados de *event sinks*. Estes *event sinks* são então associados aos eventos disponíveis no *MS Exchange Server* e às caixas postais dos usuários. Dessa forma, é possível interceptar eventos e associá-los às caixas postais de usuários pré-definidos.

Para permitir a manipulação dos dados contidos na mensagem, assim como a inclusão de uma marca para registro do identificador do *thread* e da mensagem pai (origem), os eventos síncronos são mais apropriados, visto que são os únicos que permitem a alteração do conteúdo das mensagens.

Os *event sinks* associados aos eventos síncronos são chamados duas vezes para cada ocorrência, no início e no final do processamento da mensagem salva ou excluída. Apenas durante a primeira fase do *event sink* as mensagens podem ter o seu conteúdo alterado. Assim, a inclusão da marca que permite a identificação do *thread* corrente e da mensagem origem precisa ser realizada nessa fase.

A Figura 3 apresenta a seqüência de disparo de eventos disponíveis no MS Exchange Server:

(1) O evento ocorre na base de dados do MS Exchange Server;

(2) Uma imagem do item é criada para permitir a sua manipulação;

(3) Métodos do evento síncrono recebem a notificação de início de operação. Nessa fase, o item pode ser modificado;

(4) A imagem do item é salva na base de dados do MS Exchange Server;

(5) Métodos do evento síncrono recebem a notificação da fase de conclusão da operação;

(6) Métodos do evento assíncrono recebem a notificação de operação.

6.2 Detalhes de implementação

A solução *ThreadMap* aponta para a necessidade de criação de uma marca na mensagem como solução para os problemas de identificação do thread e da mensagem origem. Para tal, foi utilizado um texto padrão, que foi inserido no final do corpo de cada mensagem interceptada no servidor, com a identificação do *thread* corrente e da mensagem pai (origem).

A Figura 4 apresenta o formato da marca inserida no corpo da mensagem.

```
.....................................
ThreadMap
{THREAD_ID:nnn}
{MSGID_ORIGEM:nnn}
.....................................
```

Figure 4. Formato da marca *ThreadMap*

As mensagens interceptadas pelo *Exchange Store Events* podem ser manipuladas através de uma fonte OLE DB chamada ExOLEDB, que juntamente com o uso do modelo de objetos *Microsoft ActiveX Data Objects* (ADO), permite o acesso às mensagens e caixas de correio eletrônico disponíveis no servidor, através da ligação (*bind*) desses itens a objetos ADO. A Tabela 3 apresenta a relação de informações colhidas sobre as mensagens interceptadas e as suas respectivas fontes.

Tabela 3. Fonte de informações das mensagens interceptadas

Informação	Fonte (URL)
Identificador da mensagem (*message_id*)	*urn:schemas:mailheader:message-id*
Emissor da mensagem (*from*)	*urn:schemas:mailheader:from*
Assunto da mensagem (*subject*)	*urn:schemas:httpmail:subject*
Data da mensagem (*date*)	*urn:schemas:mailheader:date*
Destinatários da mensagem (*to e cc*)	*urn:schemas:mailheader:to* *urn:schemas:mailheader:cc*
Existência de anexos na mensagem (*attachment*)	*urn:schemas:httpmail:hasattachment*
Corpo da mensagem	*urn:schemas:httpmail:textdescription* *urn:schemas:httpmail:htmldescription*

Com o uso de objetos ADO ligados às mensagens e caixas postais no servidor de correio eletrônico, a sua manipulação pode ser realizada através das referências URL: *"urn:schemas:mailheader"* e *"urn:schemas:httpmail"*. Assim, é possível acessar os dados da mensagem para registro das informações na base de dados do *ThreadMap* e inclusão da marca de identificação do *thread* e mensagem origem.

6.3 Interface de visualização - *ThreadViewer*

O objetivo da ferramenta *ThreadViewer* é apresentar uma interface que permita aos seus usuários visualizar os *thread* mapeados com o *ThreadMap*, possibilitando a localização de um *thread* específico e posterior visualização de seus detalhes.

Diferentemente das interfaces discutidas anteriormente, o *ThreadViewer* não se propõe a ajudar o usuário de correio eletrônico a interagir com a conversação em curso. A intenção é facilitar a busca por *threads* mapeados, a partir de suas características, tendo sido criada apenas para permitir a visualização dos *threads* mapeados pelo *ThreadMap*, tendo apenas algumas funcionalidades para análise visual de *threads*.

Para facilitar esse processo de análise, foi mapeado visualmente o seguinte conjunto de características básicas do *thread*, coletados pelo *ThreadMap*: quantidade de usuários participantes do *thread*, quantidade mensagens trocadas, duração, usuário que iniciou o *thread* e a sua data de início

A Tabela 4 apresenta um resumo dessas características e descreve a fonte dessas informações. Vale repetir que todos esses dados estão disponíveis na base do *ThreadMap*.

Tabela 4. Características dos *threads*

Característica do *thread*	Fonte da informação
Quantidade de participantes	Quantidade de usuários envolvidos no *thread*, obtida a partir das informações de emissor e receptores da mensagem (*to e cc*).
Quantidade de mensagens	Quantidade de mensagens trocadas e mapeadas pertencentes ao *thread*.
Duração do *thread*	Duração do *thread*, obtido através da diferença de dados entre a primeira e última mensagem mapeada do *thread*.
Usuário início	Usuário que iniciou o *thread*, ou seja, o emissor da primeira mensagem do *thread* mapeado.
Início do *thread*	Data em que se iniciou o *thread*, ou seja, a data da primeira mensagem do *thread* mapeado.

Para apresentar uma interface de localização mais amigável, os *threads* são apresentados em um gráfico e foram escolhidas três dimensões principais para a busca de um *thread*, são elas: a duração, a quantidade de participantes e quantidade de mensagens.

A Figura 5 apresenta a interface principal do *ThreadViewer*, com os *threads* desenhados em formato de círculos, em um gráfico tendo a quantidade de mensagens e de participantes como eixos X e Y, respectivamente. A duração representa a terceira dimensão do gráfico e está ilustrada pelo tamanho dos círculos no gráfico.

Ao lado do gráfico com os *threads* desenhados, é disponibilizado uma paleta com cinco controles que atuam dinamicamente na seleção dos *threads* à medida que são manipulados. Esses controles filtram o conjunto de dados consultados, conforme descrito a seguir:

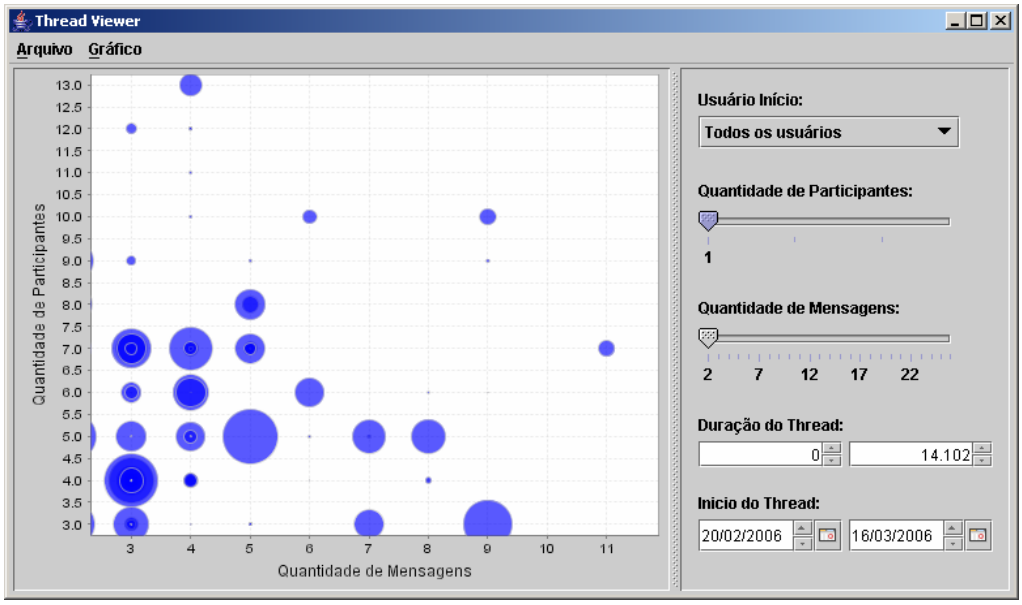


Figure 5. Interface principal do *ThreadViewer*

• Usuário início – Seleção de um usuário que tenha iniciado o *thread*;

• Quantidade de participantes – Seleciona a quantidade de participantes do *thread*. Os *threads* mostrados terão quantidade de participantes igual ou superior ao valor selecionado;

• Quantidade de mensagens – Seleciona a quantidade de mensagens do *thread*. Os *threads* mostrados terão a quantidade de mensagens igual ou superior ao valor selecionado;

• Duração do *thread* – Seleciona o período de duração dos *threads*;

• Início do *thread* – Seleciona os *threads* que se iniciaram dentro de um determinado intervalo de tempo.

Após a localização de *threads* de interesse específico, o usuário pode visualizar os seus detalhes de um thread específico clicando sobre o mesmo.

A Figura 6 apresenta a interface de detalhes do *thread*, dividida em três partes: na parte superior, os dados do *thread* selecionado (assunto, usuário início, duração, quantidade de participantes e data de início e fim); no lado esquerdo temos uma árvore montada a partir das mensagens trocadas com os respectivos relacionamentos de pai e filho; e à direita, os detalhes da mensagem selecionada à esquerda (assunto, usuário emissor, usuários receptores, data da mensagem e se houve ou não arquivos em anexo).

A ferramenta *ThreadViewer* foi construída utilizando a linguagem Java, acessando a base de dados do *ThreadMap* via *Java Database Connectivity* (JDBC). Também foram utilizados os *frameworks JFreeChart e JUNG* (*Java Universal Network/Graph*) para auxiliar no processo de criação dos gráficos necessários à visualização dos *threads* e suas respectivas mensagens.

7. ESTUDO DE CASO

O estudo de caso apresentado a seguir reflete a utilização da solução *ThreadMap* para a interceptação e mapeamento dos *threads* de mensagens eletrônicas envolvendo a equipe da Gerência de Tecnologia (GETEC) da Diretoria de Tecnologia de Informação (DTI) da Secretaria da Fazenda do Estado da Bahia.

O processo de coleta e mapeamento dos *threads* foi realizado durante o período de 20 de fevereiro de 2006 a 16 de março de 2006, envolvendo os 19 profissionais da GETEC. Durante esse período foram interceptadas e registradas em banco de dados 7.902 mensagens recebidas e trocadas entre os envolvidos no estudo de caso.

7.1 Protocolo de análise

A análise dos dados coletados foi realizada utilizando como premissa a não identificação individual dos participantes, assim como também o não tratamento dos assuntos das mensagens trocadas durante o período da coleta dos dados.

Dessa forma, os dados apresentados buscam caracterizar o uso do correio eletrônico neste grupo de trabalho e identificação do volume de mensagens trocadas, *threads* mapeados e suas características. Para tal, foram consideradas três dimensões do *thread*: quantidade de indivíduos participantes, quantidade de mensagens trocadas e duração.

Foram desconsiderados como *threads* a serem analisados, aqueles que tiveram apenas uma mensagem trocada, o que representa apenas o ato de informar ou divulgar algo, não caracterizando o uso do correio eletrônico como mecanismo de discussão.

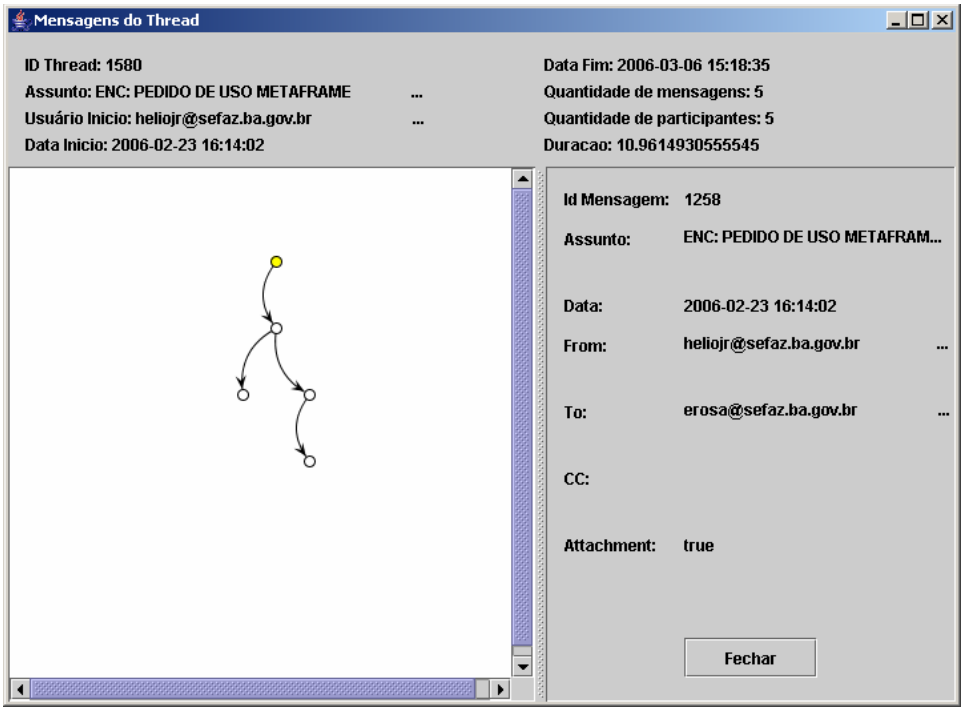


Figure 6. Visualização das mensagens do *thread* no *ThreadViewer*

Os *threads* analisados foram agrupados de acordo com a distribuição apresentada na Tabela 5, em cada uma das dimensões verificadas.

Tabela 5. Dimensões e agrupamento utilizados para análise dos dados

Dimensão	Agrupamento
Quantidade de mensagens	2 mensagens
	3 a 5 mensagens
	6 a 8 mensagens
	9 a 12 mensagens
	Acima de 12 mensagens
Quantidade de participantes	1 a 4 participantes
	5 a 8 participantes
	9 a 12 participantes
	13 a 16 participantes
	Acima de 16 participantes
Duração	Menor ou igual a 1 dia
	Entre 1 e 3 dias
	Entre 3 e 5 dias
	Entre 5 e 7 dias
	Superior a 7 dias

7.2 Resultados obtidos

Durante o período de realização do estudo de caso da GETEC, foram interceptadas 7.902 mensagens, agrupadas em 7.288 "*threads*". Como para efeito de análise dos resultados obtidos foram considerados apenas os *threads* com mais de uma mensagem, esse universo se resumiu a 261 *threads*, com um total de 889 mensagens, ou seja, 11,25% das mensagens interceptadas e registradas na base de dados do *ThreadMap*.

Este grande volume de *threads* com apenas uma mensagem pode ser atribuído ao grande volume de mensagens spam recebidas e pelo fato da equipe GETEC utilizar uma ferramenta de monitoramente e gerenciamento dos servidores e serviços disponíveis que enviam alertas via correio eletrônico.

A seguir, iremos detalhar os números obtidos neste estudo de caso, consolidando a análise a partir das características: quantidade de mensagens, quantidade de participantes e duração dos *threads*.

• Quantidade de Mensagens

Dentre os 261 *threads* analisados, houve o registro de 889 mensagens trocadas, o que perfaz uma média de 3,4 mensagens por *thread*.

A Figura 7 apresenta a distribuição de *threads* de acordo com o agrupamento de quantidade de mensagens, conforme proposto na Tabela 5.

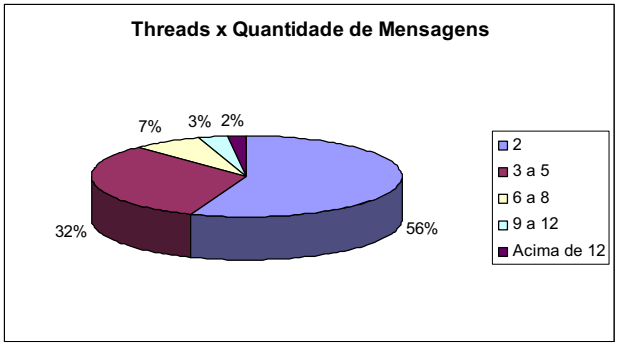


Figure 7. Distribuição de *threads* x mensagens

• Quantidade de Participantes

Para a análise da quantidade de participantes de cada *thread*, foram considerados os envolvidos como emissor e receptor de mensagens, através dos campos *From, To e CC*, contando a sua ocorrência uma única vez, em todo o *thread*.

O número total de participantes nos *threads* coletados foi de 305 pessoas, com uma média de 5,5 por *thread*. Vale lembrar que apesar do estudo de caso englobar a interceptação de mensagens de um grupo de 19 pessoas, esses participantes aqui referenciados incluem outras pessoas que enviaram e/ou receberam estas mensagens.

A Figura 8 apresenta a distribuição de *threads* de acordo com o agrupamento de quantidade de participantes, conforme proposto na Tabela 5.

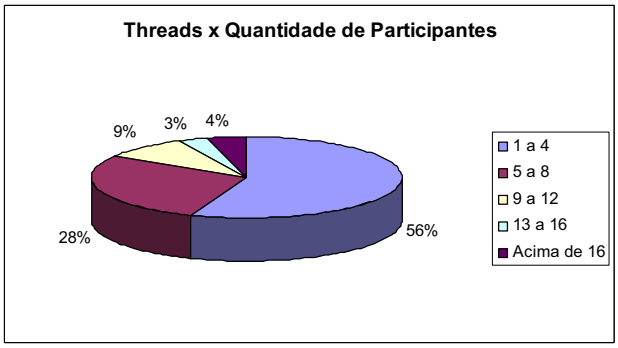


Figure 8. Distribuição de *threads* x participantes

• Duração

Para a análise da duração, foram consideradas as datas das mensagens mais antigas e mais recentes de cada *thread*. Desta forma, a duração foi calculada a partir da diferença entre a maior e a menor data das mensagens de um *thread*, contada em dias.

De acordo com os dados coletados, a duração média entre os 261 *threads* mapeados foi de 0,95 dias.

A Figura 9 apresenta a distribuição de *threads* de acordo com o agrupamento de duração, conforme proposto na Tabela 5.

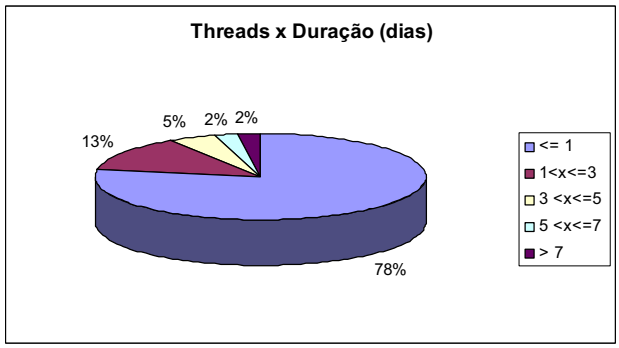


Figure 9. Distribuição de *threads* x duração

Segundo os dados coletados, dentre os 261 *threads* analisados, 84, ou seja, 32% destes foram realizados com a troca de apenas duas mensagens, a participação de até quatro pessoas e duração máxima de um dia, caracterizando assim este como o *thread* típico da equipe GETEC.

8. CONCLUSÃO

O correio eletrônico é uma ferramenta bastante difundida entre os usuários consumidores de tecnologia da informação e o seu uso representa uma importante fonte de informações.

Uma das utilizações comuns do correio eletrônico é possibilitar a integração de pessoas geograficamente distribuídas, encurtando as distâncias e facilitando a disseminação e troca de informações entre grupos de trabalho ou pessoas interessadas em assuntos em comum. Esta interação propicia a realização de conversações de uma forma virtual, entre indivíduos geograficamente espalhados, sem a necessidade de que todos estejam disponíveis ao mesmo tempo, pelo caráter assíncrono desta ferramenta.

Estas conversações através do correio eletrônico, uma vez mapeadas, permitem a montagem de uma base de dados que auxilia as organizações a entenderem o uso desta ferramenta por suas equipes de trabalho e permite a identificação de fluxos de informação ainda não conhecidos.

A solução *ThreadMap* traz as seguintes contribuições:

• Trata-se de uma proposta de mapeamento baseada na interceptação de mensagens no servidor de correio eletrônico, minimizando a possibilidade de perda de trechos destes *threads*;

• Realiza um mapeamento que possibilita a visão global do *thread*, ao contrário das soluções de mapeamento que utilizam as caixas postais dos usuários, que apresentam uma visão limitada a um dos envolvidos na conversação;

• A solução *ThreadMap* pode ser implementada em qualquer servidor de correio eletrônico que permita a interceptação e manipulação das mensagens trocadas entre os usuários.

Como o conceito do *ThreadMap* está baseado da possibilidade de interceptação de mensagens no servidor de correio eletrônico, a disponibilidade desta captura e o acesso às mensagens são limitações desta solução.

Outra limitação está no fato de que o mapeamento precisa ser realizado no momento em que as mensagens são trocadas entre os usuários, não sendo possível a realização deste mapeamento posteriormente, como são realizados pelas técnicas que utilizam como fonte de informações, as caixas postais dos usuários.

Maiores detalhes deste trabalho podem ser encontrados em [9].

9. REFERÊNCIAS

[1] Fisher, D.; Moody, P. *Studies of automated collection of email records*: technical report UCI-ISR-02-4. Institute of Software Research – University of California, Irvine, 2002.

[2] IMAP Extension Work Group. *Internet message access protocol: sort and thread extensions*. 2004. Disponível em: < http://www.ietf.org/internet-drafts/draft-ietf-imapext-sort-17.txt>. Acesso em: 5 jan. 2006.

[3] Kerr, B. *Thread Arcs: An Email Thread Visualization*. In: IEEE Symposium on Information Visualization, 2003, Seattle.

[4] Lewis, D.; Knowles K. *Threading electronic mail: a preliminary study*. Information Processing and Management, mar. 1997.

[5] Microsoft. Microsoft Exchange SDK Development Tools. 2005. Disponível em: <http://www.microsoft.com/>. Acesso em: 10 mai. 2005.

[6] Nenkova A.; Bagga, A. *Facilitating Email Thread Access by Extractive Summary Generation*. In: International Conference Recent Advances in Natural Language Processing, 2003, Borovets.

[7] Newman, P. *Treetables and other visualizations for email threads*. Xerox PARC Technical Report, September 2001.

[8] Palme, J. *Message Threading in E-mail Software*, 1998. Disponível em: <http://dsv.su.se/jpalme/ietf/message-threading.pdf>. Acesso em: 12 ago. 2004.

[9] Queiroz Junior, H. *ThreadMap – Uma ferramenta para mapeamento e visualização de threads de mensagens eletrônicas*. 2006. 105f. Dissertação (Mestrado em Redes de Computadores). UNIFACS, Salvador.

[10] Rohall, S. *Redesigning Email for the 21th Century Wokshop Position Paper*. In: Conference on Computer Supported Cooperative Work, 2002, New Orleans.

[11] Rohall, S.; Gruen, D. Remail: *A Reinvented Email Prototype*. In: Conference on Computer Supported Cooperative Work, 2002, New Orleans.

[12] Venolia, G.; Neustaedter, C. *Understanding Sequence and Reply Relationships within Email Conversations: a mixed-model visualization*. In: Conference on Human Factors in Computing System, 2003, Fort Lauderdale.

[13] Wasserman, S.; Faust, K. *Social Network Analysis: Methods and Applications*, 1994. Cambridge University Press.

[14] Zawinski, J. *Message Threading*, 1997. Disponível em: <http://www.jwz.org/doc/threading.html>. Acesso em: 20 ago. 2004.

NegoSys: um ambiente de suporte à negociação

Melise Maria Veiga de Paula
Faculdade Metodista Granbery
Programa de Engenharia de Sistemas e Computação (COPPE)
Universidade Federal do Rio de Janeiro, Brasil
mel@cos.ufrj.br

Jano Moreira de Souza
Programa de Engenharia de Sistemas e Computação (COPPE)
Instituto de Matemática
Universidade Federal do Rio de Janeiro, Brasil
jano@cos.ufrj.br

RESUMO

As constantes alterações no cenário social, econômico e político têm intensificado as atividades de negociação nas organizações e ampliado o interesse de pesquisadores neste processo. As tecnologias apresentadas na literatura sobre sistemas de suporte a negociação apresentam grandes possibilidades para troca de informação, automatização e suporte à tomada de decisão na negociação. Entretanto, a despeito do reconhecimento da necessidade e valorização da gestão do conhecimento, pouco esforço tem sido realizado no sentido de utilizá-la para capturar e gerir o conhecimento adquirido durante a negociação. Portanto, o objetivo deste trabalho é a definição e desenvolvimento de um ambiente de negociação eletrônica que combina um modelo de negociação e um modelo de gestão do conhecimento. Este ambiente deverá ser capaz de gerenciar o conhecimento adquirido em cada negociação, facilitar a aquisição de novos conhecimentos, criar meios eficientes de interação entre negociadores aliados para que eles troquem experiências e disseminem o conhecimento adquirido e entre negociadores oponentes, facilitando a troca de informações, a comunicação e uma possível cooperação entre eles.

ABSTRACT

The constant changes of the social, economic and political scenario have intensified negotiation activities in organizations and increased academic interest in the process. The technologies presented in the literature on negotiation support systems hold huge possibilities for the exchange of information, automatization, and support to decision-making in the negotiation. However, despite the acknowledgment of the need, and the value attached to the management of knowledge, little has been done in the sense of using it to capture and manage the knowledge acquired during the negotiation. The objective of this work is, thus, the definition and development of an environment of electronic negotiation that blends a negotiation model and a knowledge management model. This environment should be capable of managing the knowledge acquired at each negotiation, facilitating the acquisition of new knowledge, creating efficient means of interaction between allied negotiators so that they can exchange experiences and disseminate acquired knowledge, and between opposing negotiators, facilitating the exchange of information, communication, and enabling a possible cooperation between them.

Palavras-Chave

Sistemas de Suporte à Negociação, Gestão do Conhecimento, CSCW.

Key-Word

Negotiation Support Systems, Knowledge Management, CSCW.

1. INTRODUÇÃO

As constantes alterações no cenário social, econômico e político têm intensificado as atividades de negociação nas organizações e ampliado o interesse de pesquisadores neste processo.

Por outro lado, a diversidade de aspectos que influenciam as decisões e os resultados obtidos em uma negociação motiva a elaboração de visões diferenciadas do processo onde são considerados diferentes instrumentos teóricos e desenvolvidas diferentes abordagens de estudo na tentativa de diminuir o efeito do conflito e facilitar a busca por acordos que maximizem os ganhos obtidos.

O caráter pessoal das negociações, a independência dos participantes na tomada de decisão e a interdependência para alcançar os objetivos contribuem para a sua complexidade. Como todo processo de tomada de decisão, um aspecto importante da negociação é a quantidade e a qualidade das informações obtidas acerca do processo. Diante da necessidade de processamento e gerenciamento dos dados, informações e conhecimento pertinentes ao processo e de uma interação eficiente entre as partes envolvidas, a aplicação correta da tecnologia se torna extremamente importante.

As tecnologias apresentadas na literatura sobre sistemas de suporte a negociação apresentam grandes possibilidades para troca de informação, automatização e suporte à tomada de decisão na negociação. Entretanto, a despeito do reconhecimento da necessidade e valorização da gestão do conhecimento, pouco esforço tem sido realizado no sentido de utilizá-la para capturar e gerir o conhecimento adquirido durante a negociação.

O objetivo deste trabalho é apresentar o NegoSys, um ambiente de suporte à negociação que foi elaborado a partir de dois modelos: um modelo de decisão e um modelo de negociação.

O modelo de decisão foi definido a partir da gestão do conhecimento, que é usada para prescrever o comportamento dos negociadores através da captura e reutilização do conhecimento

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 129-139.

adquirido durante a negociação. Já o modelo de negociação do NegoSys foi baseado em uma adaptação do Modelo 3C.

O artigo está organizado como se segue: na seção 2, são apresentados alguns aspectos teóricos negociação. Na seção 3, é apresentada a arquitetura do NegoSys, bem como, algumas considerações sobre os modelos elaborados. Na seção 4, é apresentado o protótipo implementado e os resultados obtidos com algumas experiências de utilização do sistema.

2. NEGOCIAÇÃO

Historicamente, a negociação vem sendo estudada por diversos pesquisadores que apresentam diferentes abordagens sobre o processo. Um aspecto que diferencia essas abordagens é a distinção entre estudo normativo, descritivo e prescritivo [1].

O foco do estudo normativo é modelar o processo assumindo um comportamento racional dos negociadores. Na abordagem prescritiva, o objetivo é prescrever o comportamento dos negociadores, de maneira que os mesmos alcancem bons resultados durante a negociação e a excelência no processo. Na abordagem descritiva, os pesquisadores estão interessados em entender e descrever o comportamento dos negociadores, com o foco nos aspectos comportamentais e na forma como as decisões são tomadas efetivamente.

Em [9], o autor apresenta uma abordagem teórica de conexão entre essas duas perspectivas que foi denominada Perspectiva Prescritiva/Descritiva. Neste trabalho, o autor descreve e analisa uma variedade de negociações realizadas em diferentes contextos. A análise dessas negociações é usada para desenvolver prescrições aos negociadores. De acordo com essa abordagem, as prescrições devem ser baseadas na melhor descrição possível do ambiente da negociação e do comportamento dos negociadores. Vários autores afirmam que esta abordagem leva a melhores prescrições do que os modelos tradicionais [1].

No trabalho apresentado por Huang [6], o autor afirma que a negociação é um processo que envolve um conjunto de entidades que se relacionam e o final do processo pode, ou não, ser um acordo. A figura 1 representa a visão do autor. As entidades são representadas pelos retângulos e os relacionamentos, pelas setas. Segundo Huang [6], uma negociação inclui um conjunto de tarefas como, por exemplo: a definição do problema, identificação e análise das alternativas, especificação das preferências, interação entre as partes e análise das concessões. Essas tarefas são conduzidas por eventos (etapas da negociação) e executadas pelas partes envolvidas na negociação. Cada negociação envolve, no mínimo, duas partes. Por outro lado, uma negociação envolve um conjunto de atributos (questões) e cada atributo determina um conjunto de alternativas e pode ser restringido por um conjunto de critérios. Por fim, cada parte age segundo uma estratégia para gerar seu conjunto de preferências em relação às alternativas referentes a cada atributo e qual a importância desses atributos.

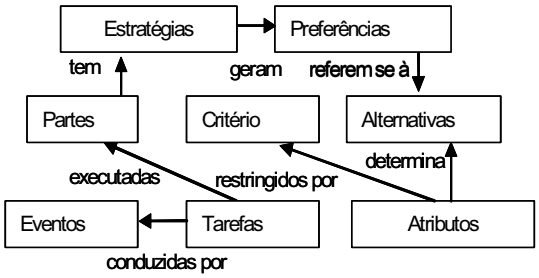


Figura 1 – Entidades e Relacionamentos da Negociação [6]

2.1 Etapas da Negociação

Na literatura sobre negociação, os autores apresentam diferentes esquematizações para o processo. Para [3] , em um processo de negociação, podem ser identificadas quatro etapas ou fases: preparação, criação de valor, divisão de valor e execução. De acordo com [2], a análise de cada uma dessas etapas, permite identificar um conjunto de elementos que determinam aspectos significativos da negociação.

A primeira etapa da negociação é a Preparação. Nesta etapa, o outro negociador pode ainda não estar presente, pois se trata do planejamento inicial. As informações relacionadas ao cenário político, econômico, social e demais fatores externos, bem como informações sobre as partes envolvidas, ajudam a identificar o contexto onde a negociação está sendo desenvolvida. Outro elemento importante é o interesse, que corresponde à relação entre a necessidade de um negociador e o objeto a ser negociado. Independente da cultura, idade, raça, cor, religião, as pessoas têm seus próprios interesses que podem variar em função das circunstâncias direcionadas pelo contexto. Para cada negociador, é essencial analisar qual é o seu real interesse e identificar os interesses da outra parte, uma vez que, para chegar a um acordo, um negociador deve proporcionar algo que desperte o interesse da outra parte sem prejudicar seu próprio interesse.

Outro aspecto importante desta etapa se refere à análise das alternativas. A razão para se negociar mostra-se presente à medida que um determinado agente (parte) detém algumas características que, quando comparadas a outros agentes (partes) com quem se poderia negociar, mostram-se mais vantajosas. Nesse sentido, quando se está inconsciente das alternativas, não há como assegurar que, daquela negociação, poderá sair o melhor acordo para uma parte ou que poderia ser melhor não fechar um determinado acordo e optar por tentar uma nova negociação. Um conceito importante para a definição dessa estrutura é a BATNA que é o acrônimo de "Best Alternative To Negotiation Agreement", conceito desenvolvido por [4]. BATNA trata-se do curso de ação baseado na preferência do negociador, caso não haja consenso. Conhecer a BATNA significa saber quais são suas alternativas se não conseguir chegar a um acordo na negociação em pauta [10].

A etapa posterior à preparação é a Criação de Valor onde os negociadores já devem estar preparados para uma maior interação com a(s) outra(s) parte (s). Deste modo, a comunicação e o relacionamento são os elementos que deverão pautar o tom das conversas.

A terceira etapa, Divisão de Valor, é por diversas vezes marcada pela contenda e pela dificuldade de comunicação, pois se trata do momento em que as partes começam a competir para a troca de

valores. Esta etapa se estende até o momento em que as partes alcançam um acordo, ou quando as partes (ou uma das) concluem que não existe nenhum acordo possível.

Assim como na Criação de Valor, a comunicação e o relacionamento entre as partes requerem uma significativa atenção nesta etapa. As partes envolvidas se comunicam e enviam suas propostas para tentar encontrar áreas de possíveis acordos. Portanto, é necessário refletir sobre a formatação dessas propostas e as possíveis concessões que possam direcionar a concordância de idéias.

Na Divisão de Valor, é fundamental que, ao apresentar suas propostas, um negociador considere os padrões analisados durante a preparação na tentativa de demonstrar como a equidade das propostas e como elas atendem às necessidades e expectativas da outra parte. Por fim, as partes devem verificar o que está sendo definido e formalizar o acordo através da elaboração do Contrato.

Na Execução, é preciso verificar o que foi acordado e tomar as medidas necessárias para o cumprimento do que foi negociado. Esta etapa envolve também as discussões sobre as garantias e mecanismos para assegurar, controlar e monitorar a execução do acordo.

Três elementos são significativos nesta etapa: o relacionamento, o tempo e a conformidade. A conformidade se refere à concordância entre o acordo obtido e as atividades a serem executadas em função deste acordo. Já a análise do tempo deve considerar as disposições estabelecidas no acordo a fim de determinar o momento apropriado para a execução dessas atividades.

2.2 Classificações da Negociação

Ao invés de definir a negociação, muitos pesquisadores e profissionais da área têm seguido um caminho que leva à descrição de dicotomias da negociação, considerando, principalmente, as formas como as decisões são tomadas e o foco dos negociadores [8].

No trabalho apresentado por [4], os autores classificam a negociação em barganha de propostas e negociação baseada em princípios. De acordo com os autores, a maioria das negociações é conduzida através da barganha posicional, que, normalmente, resulta em impasses difíceis de solucionar ou em um acordo que é percebido por uma das partes como tendo sido imposto devido à força superior da outra parte. Os autores sugerem que em vez de se basear em posições, a negociação deve ser baseada em princípios e deve ter como ênfase os interesses subjacentes que motivam as partes a assumirem suas posições. Dessa forma, podem ser desenvolvidas soluções criativas que atendam, pelo menos em parte, aos interesses subjacentes de cada uma das partes, permitindo assim uma resolução do conflito baseada em princípios e mutuamente vantajosa.

Outra classificação da negociação encontrada na literatura é baseada na estratégia adotada pelos negociadores. Na definição da estratégia, podem existir alguns dilemas, como por exemplo: cooperação ou competição e curto prazo ou longo prazo. A resposta para esses dilemas define o tipo da negociação. Tradicionalmente, existem dois tipos de negociação: competitiva (distributiva) e cooperativa (integrativa) [9]. A negociação competitiva ou distributiva (também conhecida como Soma-Zero no contexto da Teoria do Jogo e Pesquisa Operacional) é classificada como Ganha/Perde (G/P). O negociador com a postura Ganha/Perde escolhe a competição e relacionamentos de curto prazo priorizando os resultados. Só opta pela verdade, caso possa tirar proveito dela e, pelo longo prazo, caso possa criar uma relação de dependência. Tem como proposição básica obter o melhor resultado possível. Deste modo, os resultados de uma parte são prejudicados em detrimento da outra.

A negociação cooperativa ou integrativa (também conhecida como negociação colaborativa) é classificada como Ganha/Ganha (G/G) sendo positiva para ambos os lados. É um processo cooperativo onde são encontradas alternativas de ganho comum, isto é, que atendam aos interesses de todas as partes. Na negociação G/G, a efetividade do acordo deve ser produto da qualidade pela aceitação, ou seja, os interesses legítimos das partes foram atendidos e as partes ficaram satisfeitas e se comprometem com o cumprimento do que foi acordado.

3. ARQUITETURA DO NEGOSYS

A partir da análise da literatura sobre negociação, foi possível identificar dois requisitos básicos para elaboração de um ambiente de negociação eletrônica: o suporte à decisão e o suporte ao processo.

O primeiro requisito aponta em direção à elaboração de um modelo de decisão capaz de capturar a lógica utilizada e o conhecimento associado à tomada de decisão durante a negociação, permitindo a posterior recuperação desses elementos.

O modelo de decisão definido no NegoSys se baseia na perspectiva Prescritiva/Descritiva proposta por [9], no qual, a gestão do conhecimento é usada para prescrever o comportamento dos negociadores através da captura e reutilização do conhecimento adquirido durante a negociação.

O segundo requisito representa a necessidade de definição de um modelo de suporte ao processo que contemple todos os componentes necessários para apoiar os negociadores na execução das atividades intrínsecas ao processo.

Considerando o aspecto social da negociação, foi possível entender o processo como uma forma específica de trabalho em grupo, na qual, o objetivo comum é alcançar um acordo mutuamente aceitável. Mais especificamente, a análise das características que fundamentam uma negociação cooperativa induz o entendimento que, neste caso, as partes trabalham cooperativamente para alcançar o acordo.

O aspecto que corrobora com esta concepção vem de uma característica singular deste processo: mesmo com a autonomia em relação à tomada de decisão, independente da estratégia utilizada e dos interesses dos negociadores, a negociação exige uma interdependência entre as partes envolvidas, uma vez que um acordo não pode ser alcançado unilateralmente. Esta interdependência representa a necessidade de uma participação conjunta das partes que precisam co-operar para que o acordo seja alcançado. Além disso, a negociação exige a comunicação entre as partes e a coordenação das atividades envolvidas.

O modelo de negociação do NegoSys foi elaborado baseado na adaptação do Modelo 3C descrita por [5]. Para negociar, é necessário trocar informações (comunicação). Para alcançar o acordo, é necessário executar algumas atividades que requerem a

operação em conjunto entre as partes (co-operação), além de uma organização dessas atividades (coordenação). A partir da co-operação e gerenciados pela coordenação das atividades, que inclui a troca de informações na comunicação, o acordo é definido e os compromissos são estabelecidos. Além disso, em um ambiente de negociação eletrônica, a percepção pode interferir diretamente em um dos elementos da negociação que foi destacado no capítulo 2, o tempo, uma vez que, a partir da percepção, os negociadores tomam ciência das atividades a serem executadas, o que pode acelerar o processo.

No que se refere à comunicação, no NegoSys, os negociadores podem se comunicar através de ferramentas que apóiam a comunicação escrita síncrona e assíncrona: mensagens instantâneas, correio eletrônico e listas de discussão.

A primeira vantagem dessa alternativa é a possibilidade de realização de diversas conversações simultâneas em paralelo, enquanto nas conversações por meio da fala, as pessoas atendem a um protocolo natural de comunicação, aguardando sua vez de falar. Outra vantagem na comunicação escrita é a possibilidade de persistência das mensagens ou anotações, facilitando aos negociadores recuperarem rapidamente uma seqüência de informação tanto da negociação a qual estão participando, quanto daquelas as quais não participaram, permitindo a reutilização dessas informações e a utilização de ferramentas de comunicação assíncronas.

Em relação à coordenação, foi necessário sistematizar a negociação e definir o fluxo de atividades a serem executadas no NegoSys. Para atender esse requisito, foi definido o **Workflow da Negociação** que, além de representar o fluxo das atividades executadas, deve ser capaz de representar a sincronização dessas atividades e o fluxo de informações. Vale ressaltar que, esta sistematização e possível automação das atividades da negociação devem ser flexíveis o suficiente para suportar as diversas variações e complexidade do processo.

O Workflow da Negociação é composto pelas seguintes etapas: Proposta de Negociação, Preparação, Definição dos Parâmetros da Negociação, Negociação dos Atributos, Contrato e Avaliação da Negociação como ilustrado na figura 2:

- proposta de negociação: fase inicial do processo de negociação onde um dos negociadores, após identificar e definir suas necessidades, deve iniciar o contato com a outra parte, enviando uma proposta de negociação;

- preparação: nesta etapa, os negociadores devem se preparar para a negociação definindo um conjunto de informações que são essenciais à tomada de decisão durante o processo, como por exemplo: atributos que considera importante, valores de reserva e nível de aspiração para cada atributo, prioridade entre os atributos especificados e análise das alternativas, interesses e opções. No NegoSys, essas informações são denominadas Parâmetros do Negociador e são usadas para o suporte à decisão. Vale ressaltar que, um negociador não tem acesso aos Parâmetros de outro negociador, essa informação não é publica;

- definição dos parâmetros da negociação: nesta etapa, os negociadores devem interagir para definição das informações comuns que serão utilizadas pelo sistema para automação de algumas atividades. Para permitir que os negociadores tenham flexibilidade em relação aos atributos negociados e à agenda da negociação, durante esta etapa, as partes envolvidas devem "negociar" quais atributos serão considerados durante o processo e os prazos estabelecidos para conclusão da negociação. Vale ressaltar que, na etapa anterior, os negociadores especificam os atributos que os mesmos consideram importantes. Entretanto, os atributos que serão realmente negociados devem ser acordados entre os negociadores durante esta etapa. Essas informações são denominadas Parâmetros da Negociação e são visualizadas por todos os negociadores envolvidos no processo. Nesta etapa, os negociadores devem interagir através de mensagens classificadas como "Proposta de Definição". Além disso, os negociadores ainda podem argumentar uma proposta enviada a partir do envio de mensagens do tipo "Argumentação", ou ainda, enviar mensagens não diretamente relacionadas a uma proposta, denominadas "Conversação";

- negociação dos atributos: nesta etapa, os atributos definidos na etapa anterior devem ser negociados de forma seqüencial. Nesta etapa, os negociadores interagem através do envio de mensagens que são classificadas como: oferta. Além disso, ainda podem enviar mensagens do tipo argumentação e conversação. É nesta etapa que os valores de cada atributo são definidos pelos negociadores. Caso o acordo seja obtido, começa a etapa denominada Contrato, onde o contrato resultante do acordo é elaborado. Entretanto, um negociador pode desistir da negociação e finalizar o processo sem acordo;

- contrato: após o acordo, o NegoSys deve gerar um contrato onde serão especificadas as seguintes informações: dados dos negociadores, o acordo obtido (atributos negociados e seus respectivos valores), datas de início e fim da negociação e observações gerais que pode ser definidas pelos negociadores;

- avaliação da negociação: nesta etapa, os negociadores apresentam suas percepções e julgamentos a respeito da negociação finalizada, onde devem ser avaliadas a negociação, o acordo obtido, a contraparte, as práticas adotadas e decisões tomadas durante a negociação. Estas informações podem ser utilizadas por negociadores de uma mesma organização para facilitar a tomada de decisão em futuras negociações.

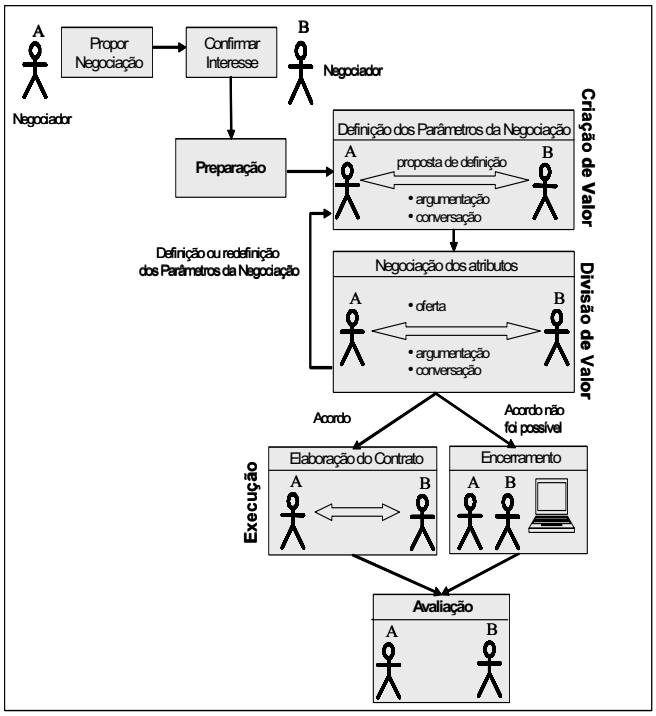


Figura 2 – Workflow da Negociação

Além do Workflow da Negociação, para utilização da tecnologia como suporte a coordenação dessas atividades, foi necessário definir o conjunto de regras que devem ser consideradas. Este conjunto de regras compõe o **Protocolo de Negociação,** definido e elaborado para o NegoSys de forma a garantir que as atividades exercidas pelos negociadores durante uma negociação sejam coerentes com o modelo proposto [8] .

No que se refere à percepção, no NegoSys, os negociadores são notificados à medida que o processo evolui. Estas notificações se referem a três aspectos diferentes: propostas e mensagens recebidas, agenda da negociação e inclusão de formas de conhecimento relevantes ao processo. Como exemplo, considere o caso onde um negociador A envia uma oferta para um negociador B. As informações referentes à oferta enviada são visualizadas somente no NegoSys. Entretanto, o sistema envia, automaticamente, uma mensagem para o correio eletrônico do negociador receptor notificando-o do recebimento da oferta.

A arquitetura do NegoSys está representada na figura 3 sendo dividida em duas camadas: i) Mesa de Negociação (Camada de Negociação) e ii) GCN (Camada de Gestão do Conhecimento na Negociação). Na Mesa de Negociação, são definidos os módulos funcionais relacionados ao suporte ao processo de negociação. No GCN, são definidos os módulos responsáveis por apoiar à gestão do conhecimento. Ainda na figura 3, é representada a Memória da Negociação que ilustra o componente da arquitetura responsável pelo armazenamento de todas as formas de conhecimento gerenciadas pelo sistema.

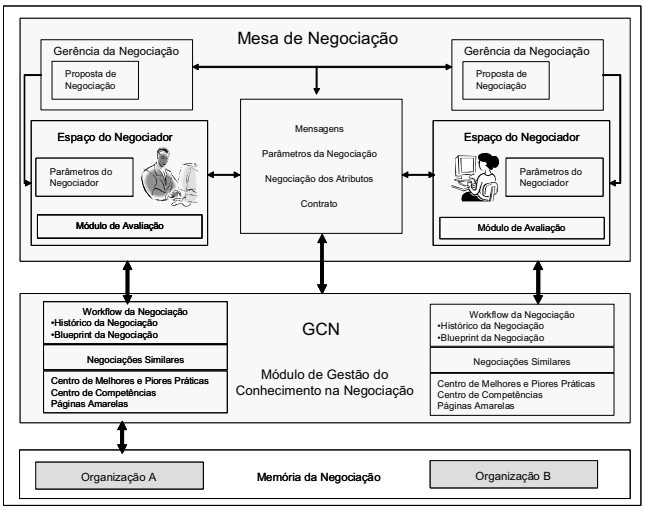


Figura 3 – Arquitetura do NegoSys

Os módulos que compõem a Mesa de Negociação são: Gerência da Negociação e Espaço do Negociador. O Gerente de Negociação de uma organização é o usuário responsável pela gerência das informações relacionadas às negociações da organização. Cabe ao gerente da negociação cadastrar os negociadores, manter as informações relacionadas à Organização e enviar propostas de negociação. A figura 4 ilustra o formulário de acesso ao módulo Gerente de Negociação.

Figura 4 – Módulo de Gerência de Negociação

O módulo Espaço do Negociador, ilustrado na figura 5, representa a área do sistema que deverá ser usada por um negociador para que o mesmo tenha acesso as demais funcionalidades do sistema, sendo composto pelos seguintes sub-módulos: Alterar Perfil, Mensagens, Negociações Finalizadas, Negociações em Andamentos e Contratos. Além disso, a partir deste módulo, o negociador tem acesso à camada de Gestão do Conhecimento (GCN).

As negociações não finalizadas, nas quais o negociador atua, são visualizadas a partir do submódulo Negociações em Andamento. Ao editar (abrir) uma negociação em andamento, o negociador visualiza algumas informações e tem acesso aos submódulos

Parâmetros do Negociador, Parâmetros da Negociação e Histórico de Propostas.

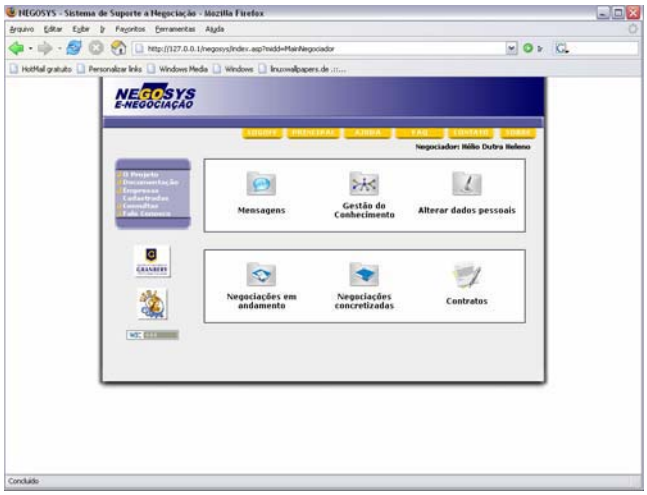


Figura 5 – Módulo Espaço do Negociador

De acordo com as etapas descritas no Workflow, durante a preparação, os negociadores devem analisar um conjunto de elementos que resulta em uma série de informações importantes para negociação atual e que podem ser extremamente úteis nas futuras negociações, facilitando a tomada de decisão. No Workflow da Negociação, essas informações foram denominadas Parâmetros do Negociador e devem ser inseridas no sistema a partir do submódulo Parâmetros do Negociador, ilustrado na figura 6.

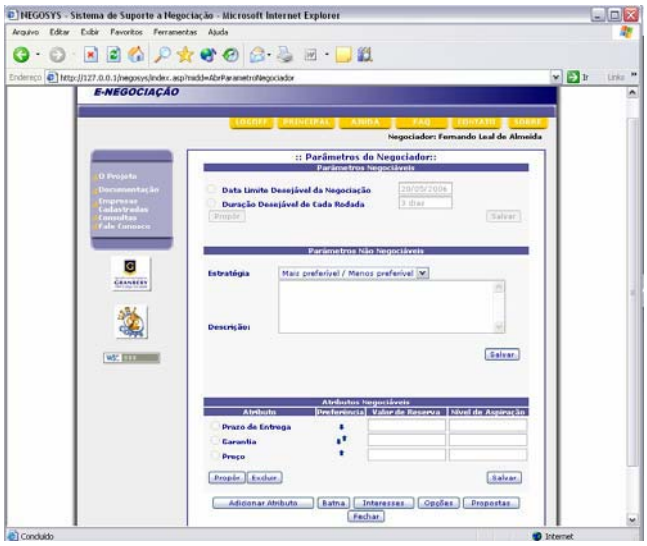


Figura 6 – Submódulo Parâmetros do Negociador

No formulário representado na figura 6, os negociadores também terão acesso às seguintes funcionalidades: elaboração da Batna, análise dos Interesses e análise das Opções.

Tendo definido os Parâmetros do Negociador, que inclui a data desejável para o término da negociação, tempo máximo desejável de cada rodada, a descrição da negociação e o conjunto de atributos negociáveis, o negociador, a partir do envio de uma proposta de definição de parâmetro, inicia a atividade de **Definição dos Parâmetros da Negociação** (descrita no Workflow). O negociador que recebe a proposta de definição de parâmetro pode optar por aceitar ou recusar a proposta.

No submódulo Parâmetros da Negociação, são visualizados todos os parâmetros que foram definidos (negociados) pelos negociadores. Portanto, ao ser enviada uma proposta de definição de parâmetro da negociação, caso o negociador receptor aceite a proposta recebida, o valor do parâmetro é definido e torna-se disponível para visualização neste submódulo, conforme ilustrado na figura 7.

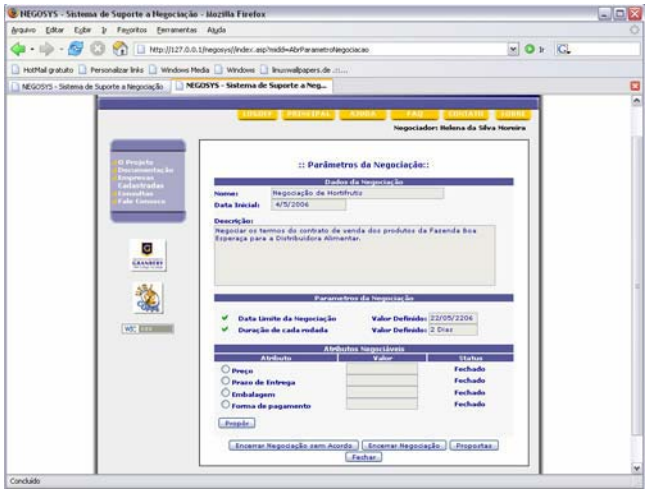


Figura 7 – Submódulo Parâmetros da Negociação

Além disso, a atividade de **Negociação de Atributos**, descrita no Workflow, é realizada a partir da interação do negociador com este submódulo quando do envio de uma mensagem do tipo oferta. Neste caso, o negociador deve propor um valor para um dos atributos listados neste submódulo, iniciando uma rodada de negociação. A figura 8 ilustra o formulário, no qual, um negociador elabora uma mensagem do tipo oferta a ser enviada à sua contraparte.

Figura 8 – Elaboração de mensagens do tipo oferta

No submódulo Elaboração do Contrato, é formalizado o contrato gerado a partir do acordo definido entre os negociadores. As informações referentes ao contrato devem ser inseridas pelo

sistema. Cabe aos negociadores aceitar o contrato ou ainda, propor a inclusão de uma observação de contrato. O módulo Avaliação é responsável por apoiar o negociador na avaliação da negociação que inclui a avaliação do acordo, contraparte, negociação e das estratégias adotadas.

No que se refere à Camada de Gestão do Conhecimento, os módulos que compõem esta camada são responsáveis por disponibilizar todas as funcionalidades necessárias à gestão do conhecimento criado no NegoSys. O conhecimento gerenciado deve ser capturado durante a interação dos agentes humanos com o NegoSys (negociador, gerente da negociação e especialistas). Sendo assim, a Mesa de Negociação pode ser considerada um dos recursos do sistema para captura do conhecimento. Por outro lado, não é o único, pois os módulos descritos no GCN poderão ser manipulados por esses agentes humanos, mesmo quando uma negociação não estiver sendo executada. Além disso, o GCN é responsável por facilitar a disseminação do conhecimento entre negociadores aliados, o que pode ser considerada uma fonte diferente para a captura do conhecimento.

Esta camada é composta pelos seguintes módulos: Workflow da Negociação, Centro de Competências, Comunidades, Negociações Similares, Centro de Piores e Melhores Práticas e Páginas Amarelas.

O Workflow da Negociação, ilustrado na figura 9, contempla duas funcionalidades do NegoSys: acesso ao histórico das atividades realizadas durante a negociação e a visualização do planejamento das atividades que devem ser realizadas durante uma negociação. As atividades definidas no modelo de negociação são representadas pelas caixas verdes. Vale destacar que, a atividade em vermelho representa a atividade em execução no momento. Cada atividade é composta de tarefas que são classificadas de acordo com o tipo e o estado da tarefa.

No que se refere ao tipo, uma tarefa é classificada como padrão quando se refere a uma das tarefas definidas nas atividades que compõe o Workflow da Negociação. Por exemplo, a elaboração do BATNA e a análise dos interesses na atividade de Preparação. Além disso, durante o andamento de uma atividade do Workflow, o negociador pode optar por incluir uma tarefa não padrão, ou seja, uma tarefa não definida no Workflow como, por exemplo, suponha que um negociador tenha consultado um especialista da organização ou tenha interagido com uma pessoa externa à organização obtendo uma informação importante para a tomada de decisão. O negociador pode inserir esta informação no sistema a partir da inclusão de uma nova tarefa. Neste caso, a informação inclui o especialista ou a pessoa com quem o negociador interagiu. Em relação ao estado, uma tarefa é classificada em: não iniciada, em andamento e concluída. O estado das tarefas classificadas como padrão é automaticamente atualizado pelo sistema. Por outro lado, o estado de uma atividade não padrão é controlado pelo negociador que é o responsável pela inclusão da tarefa.

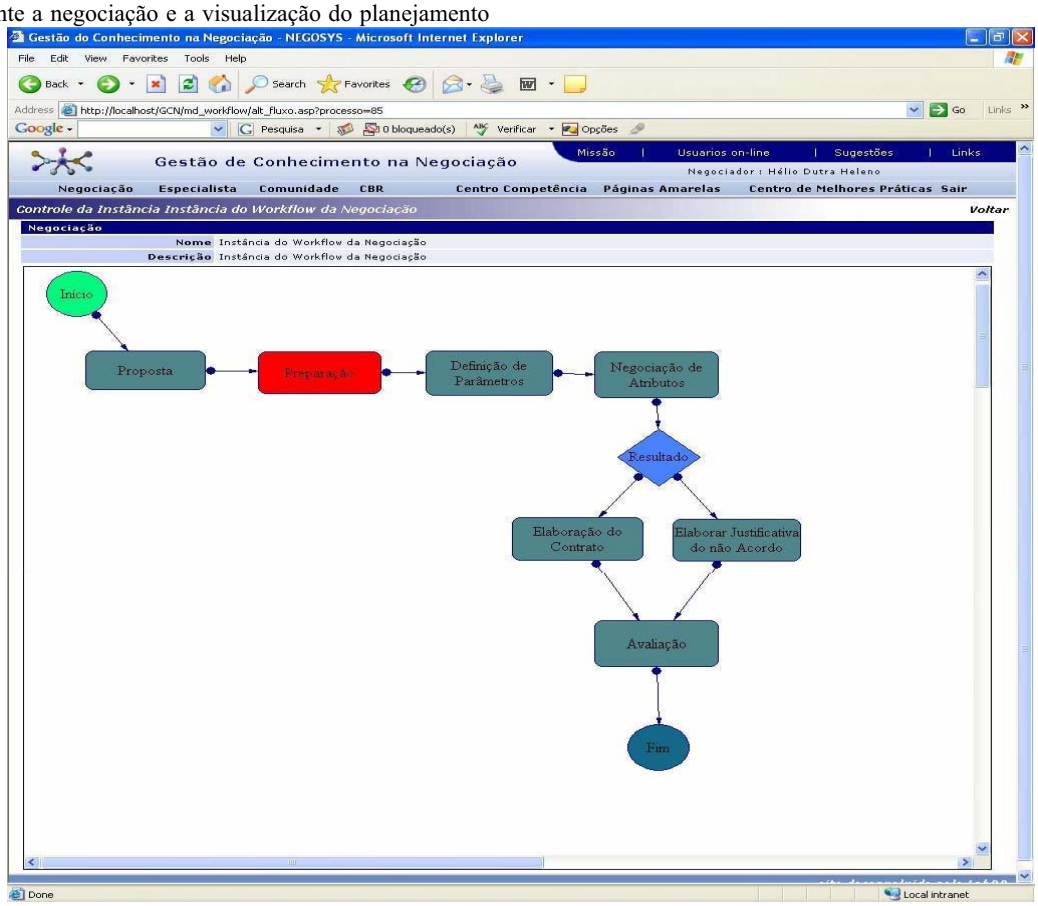


Figura 9 – Módulo Workflow da Negociação

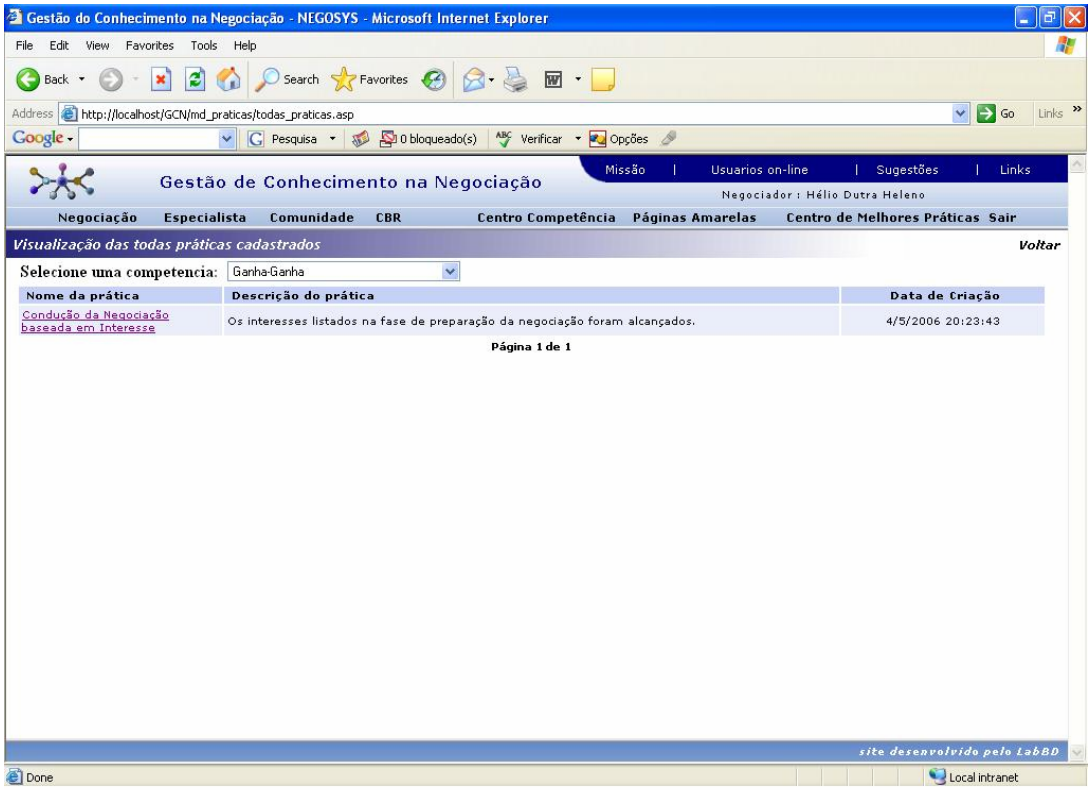


Figura 10 – Módulo Centro de Melhores e Piores Práticas

No módulo Centro de Melhores e Piores Práticas, ilustrado na figura 10, os negociadores poderão disponibilizar informações sobre as práticas adotadas em uma negociação e seus resultados. Vale ressaltar que, em uma negociação, os erros também são fontes importantes de aprendizado e uma experiência mal sucedida deve ser armazenada e documentada para que se evitem futuros erros. Por isto, neste módulo, os negociadores também podem cadastrar práticas de negociações mal-sucedidas.

No NegoSys, toda prática deve ser associada às seguintes informações: a negociação que originou a prática, o autor da prática, a identificação da contraparte que participou da negociação, a pratica adotada e, caso seja uma prática de sucesso, que melhorias esta acarretou à negociação e os detalhes sobre o que foi obtido com a prática como, por exemplo, inovação e melhoria de estratégias, redução do tempo de negociação, otimização dos resultados e melhorias no relacionamento com a outra parte. Por outro lado, caso a prática não tenha obtido bons resultados, os problemas gerados devem ser especificados.

Cada prática deve ser associada a um termo-chave que é usado para descrever certos aspectos da realidade e que tem um significado para negociação, facilitando o acesso e navegação pela base de conhecimento. No NegoSys, os termos que qualificam os itens de conhecimento criados e disponibilizados foram definidos em um Vocabulário de Negociação elaborado a partir da pesquisa da literatura do processo. Os termos definidos no vocabulário são organizados em áreas de competências. Além disso, cada área está associada a um conjunto de competências. Mais detalhes sobre o Vocabulário da Negociação pode encontrado em PAULA (2006).

No NegoSys, o módulo Comunidades provê um fórum para troca e aquisição do conhecimento tácito referente a um domínio, permitindo que negociadores aliados interajam e troquem conhecimento. Essa troca e aquisição do conhecimento podem ser de maneira síncrona ou assíncrona. Na forma síncrona, os negociadores podem entrevistar e realizar reuniões virtuais com um especialista através da utilização de ferramentas de reunião eletrônica. Ou então, uma dúvida é submetida a uma Comunidade, e algum especialista pertencente a esta comunidade pode respondê-la posteriormente. Esta é a troca assíncrona de conhecimento.

O módulo Centro de Competências é responsável por permitir a identificação das competências disponíveis a uma organização. A partir desta associação, é possível identificar profissionais dentro da organização, outros negociadores ou não, que podem ser consultados quando um especialista é necessário para executar uma tarefa ou resolver um problema ou ainda, encontrar um negociador com um determinado conhecimento quando o responsável pela negociação é afastado. Entretanto, para prover uma identificação do conhecimento, no NegoSys, é possível executar as seguintes atividades: associar especialistas a competências; associar as competências com o conhecimento explícito (documentos que representam uma competência) e associar uma negociação a competências. A figura 11 ilustra o

formulário, no qual, é possível consultar um especialista associado a uma determinada competência.

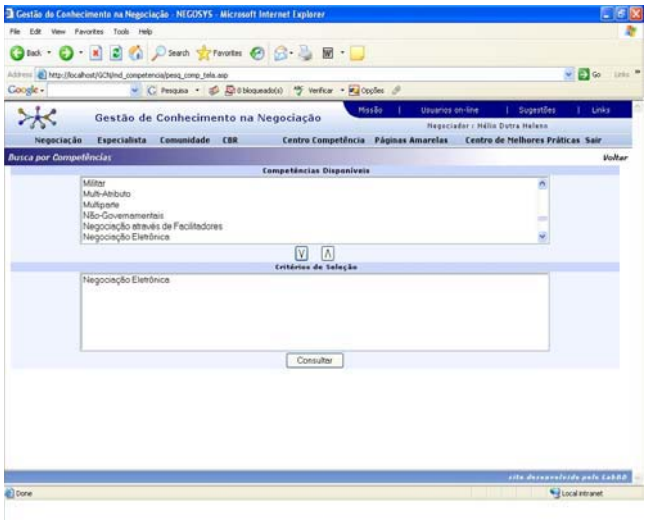


Figura 11 – Consulta por especialista no Centro de Competências

O módulo de Páginas Amarelas pode ser usado para localizar facilmente fornecedores de dados, bem como realizar um controle da qualidade e periodicidade do dado fornecido.

Para isto, este módulo deve dispor de ferramentas de cadastro, alteração, edição e busca de fornecedores de dados, periodicidade que o dado é fornecido, as negociações que utilizaram esses dados e informações referentes à avaliação do dado. A figura 12 ilustra o formulário que exibe as informações de um dado cadastrado no sistema.

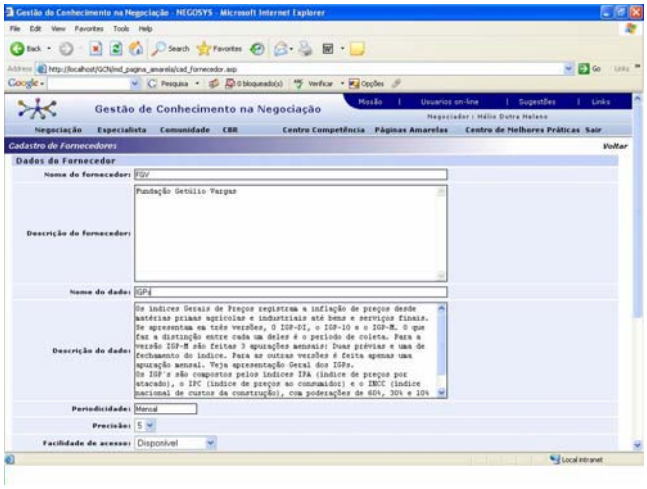


Figura 12 – Módulo Páginas Amarelas

O módulo de Negociações Similares deve permitir a consulta em negociações anteriores de maneira que o conhecimento criado possa ser reutilizado para facilitar as atividades do negociador durante a especificação dessas informações. Para realização dessa função, no NegoSys, os negociadores podem realizar consultas na base de negociações baseadas nos seguintes critérios: nome da organização que participou da negociação; nome da contraparte; atributos considerados; condição de término da negociação: sem ou com acordo e data final e inicial da negociação.

4. PROTÓTIPO

Nesta seção, serão apresentadas algumas considerações sobre o protótipo do NegoSys que foi implementado a partir da arquitetura descrita na seção anterior, desenvolvido a partir das seguintes especificações de software: Microsoft Internet Information Services – IIS (Servidor Web), Microsoft SQL Server 2000 (Servidor de Banco de Dados), Macromedia Dreamweaver (Editor HTML/ASP) e Macromedia Fireworks (Editora de Imagens).

As folhas de estilos (CSS) e todo o código HTML foram desenvolvidos de acordo com o padrão definido pelo W3C. Além disso, a organização do código gerado foi estruturada baseada no padrão de projeto MVC (Model View Controller), dividindo o código em 3 camadas que foram armazenadas em diretórios diferentes: classes de domínio, visões e controladores. Esta estrutura facilitou o processo de desenvolvimento do protótipo, isolando as regra de negócio da interface.

4.1 Experimentos Realizados

No atual estágio de desenvolvimento do NegoSys, foi realizada uma experiência de utilização do sistema a partir da simulação de um caso fictício de negociação. Nesta experiência, o NegoSys foi posto em prática na tentativa de validar, principalmente, a hipótese de que o modelo de negociação elaborado, que foi usado como base para definição dos componentes de suporte ao processo, representa uma alternativa viável para efetivar um processo de negociação. A seguir, serão descritos os recursos utilizados, o cenário, os participantes e as atividades realizadas durante as experiências.

- Recursos utilizados: um questionário para qualificação dos participantes; um questionário de avaliação do NegoSys; um documento contendo a descrição do problema (caso de negociação elaborado); um documento contendo algumas considerações a respeito da utilização do Negosys;
- Cenário: foram realizadas 8 negociações considerando o mesmo caso. Além disso, os negociadores que representaram as organizações definidas no caso de negociação foram dispostos em salas diferentes de acordo com a parte que representavam. Foi solicitado que, durante as negociações, os participantes que representavam partes oponentes não se comunicassem, a não ser, a partir do NegoSys.
- Participantes: foram convidados 8 participantes, entre eles, dois alunos de doutorado e seis alunos de mestrado do Programa de Engenharia de Sistemas da COPPE. Algumas informações a respeito da experiência desses participantes como negociadores e de utilização de sistemas de suporte à negociação foram identificadas a partir do questionário de avaliação. Todos os participantes informaram não possuir experiência na utilização de sistemas de suporte à negociação. Três dos oito participantes possuem algum conhecimento sobre o processo de negociação a partir da análise de alguns trabalhos da literatura.
- As etapas realizadas no procedimento foram:
  - Preenchimento do questionário de qualificação;

- Leitura, por parte de cada participante, da descrição da negociação, considerando as informações particulares;
- Explicação oral sobre o uso do NegoSys;
- Negociação usando o NegoSys;
- Preenchimento do questionário de avaliação do NegoSys

A partir destes experimentos inicias, foram identificadas informações importantes para o aprimoramento do sistema. No que se refere à facilidade de utilização do NegoSys, basicamente, os participantes não apresentaram muita dificuldade. Uma observação importante se refere à utilidade do Workflow da Negociação que foi considerado pelos participantes um facilitador para a utilização do sistema, uma vez, que o fluxo de tarefas e a descrição de cada uma delas estão acessíveis a qualquer momento.

Em relação ao modelo de negociação elaborado, pode-se afirmar que a hipótese mencionada anteriormente foi confirmada, visto que, a representação da negociação e o fluxo de atividades definidas na fase de preparação foram considerados, pela maioria dos participantes, como sendo de extrema utilidade. Portanto, pode-se concluir que os componentes de suporte ao processo definidos no NegoSys representam uma alternativa viável para efetivação da negociação. Contudo, foram identificadas novas funcionalidades que podem facilitar esta preparação. Estas funcionalidades foram traduzidas em novos requisitos funcionais que vem sendo implementados no sistema.

5. CONSIDERAÇÕES FINAIS

Este estudo procurou investigar a negociação e todos os requisitos necessários para elaboração de um ambiente computacional na WEB que permita que negociadores utilizem este ambiente e aprimorem sua capacidade de negociar a partir da reutilização do conhecimento que é criado durante o processo.

As principais contribuições deste trabalho se referem à elaboração do modelo de negociação e do modelo de gestão do conhecimento na negociação.

O modelo de negociação foi elaborado de forma a garantir que:

- As atividades dos negociadores quando da utilização do NegoSys facilitem o máximo reaproveitamento do conhecimento gerado durante o processo.
- A colaboração entre negociadores aliados e a cooperação entre negociadores oponentes seja efetiva e eficiente.

As contribuições deste modelo podem ser categorizadas em função de quatro aspectos: comunicação, coordenação, cooperação e percepção.

- Comunicação: Assim como nos outros SSN analisados, algumas ferramentas para o suporte ao trabalho em grupo foram utilizadas. No NegoSys, a utilização destas ferramentas foi associada às idéias advindas da Teoria dos Atos de Fala, permitindo a comunicação escrita entre os negociadores oponentes, que é qualificada de acordo com a intenção dos mesmos. Contudo, o grande diferencial em relação aos demais trabalhos é o suporte à socialização do conhecimento a partir da colaboração entre os negociadores aliados.
- Coordenação: assim como nos outros SSN analisados, no NegoSys, foi elaborado um protocolo de negociação para garantir que as atividades executadas pelos negociadores sejam coerentes com o modelo de negociação proposto. Contudo, o protocolo de negociação do NegoSys, se comparado aos outros trabalhos, é mais flexível em relação aos seguintes aspectos:
    - Flexibilidade em relação à escolha dos atributos considerados
    - Flexibilidade em relação ao momento de escolha dos atributos (os atributos podem ser determinados antes ou durante a fase de negociação dos atributos);
    - Flexibilidade em relação ao cronograma da negociação;
    - Re-negociação de atributos e do cronograma
    - Negociação para a inclusão de observação no contrato

Ainda no que se refere à coordenação, o Workflow da Negociação representa três diferencias significativos em relação aos demais trabalhos. Primeiro, no NegoSys, é possível visualizar o histórico de todas as atividades realizadas, não somente das propostas enviadas e recebidas. Segundo, no NegoSys, é possível que os negociadores façam o registro de outras atividades diferentes das que foram definidas no Workflow, como por exemplo, a consulta de especialistas e documentos. Por fim, no NegoSys, ainda é possível planejar as atividades futuras.

- Cooperação: a operação em conjunto entre negociadores oponentes é uma característica comum entre o NegoSys e os demais trabalhos analisados. Contudo, no NegoSys, este aspecto é estendido a partir do suporte á colaboração entre negociadores aliados.
- Percepção: assim como nos demais trabalhos, a percepção no NegoSys é apoiada pelo envio de notificações. Contudo, no NegoSys, a inclusão de novas fontes de conhecimento também é notificada, característica que o diferencia dos demais trabalhos.

Ainda em relação ao modelo de negociação, a confiabilidade das informações foi garantida a partir dos seguintes recursos:

- Classificação das informações em *Parâmetros do Negociador* e *Parâmetros da Negociação* e o acesso exclusivo dos negociadores de uma determinada organização àquelas informações descritas como parâmetros do negociador
- Acesso à Memória da Negociação somente com a permissão do Gerenciador de Negociação
- Acesso ao ambiente através de senha

No que tange as contribuições do modelo de gestão do conhecimento, destaca-se a identificação das diversas formas de conhecimento criadas durante negociação, o que permitiu definir componentes na arquitetura do NegoSys que garantissem a

identificação, captura e armazenamento dessas formas de conhecimento. Além da identificação das possíveis associações entre situações de problemas semelhantes para reutilização das formas de conhecimento disponíveis.

REFERÊNCIAS

[1] Bell, D. E., Raiffa, H. and Tversky, A. *Decision making: Descriptive, normative, and prescriptive interactions*. Cambridge, Eng.: Cambridge University Press, 1989.

[2] Duzert, Y, Paula, M.M.V., Souza, J. M. *Case study on complex negotiation analysis*. Relatório Técnico – Programa de Engenharia de Sistema e Computação, Universidade Federal do Rio de Janeiro, Versão estendida submetida ao Negotiation Journal, janeiro/2006. Disponível em: www.cos.ufrj.br.

[3] Duzert, Y., Ruediger, M.A., Riccio, V, Bulhões, F. Concilier le Changement et la Contruction de Consensus: le cas des réformes au Brésil. *Revue Française de Gestion*, v. 30, n.153, pp. 173-184. 2005.

[4] Fisher, R., Ury, W. *Getting to Yes: Negotiating Agreement without Giving In*, 1 ed, New York, Penguin Books. 1981.

[5] Fuks, H., Gerosa, M.A., Raposo, A.B., Lucena, C.J.P. O Modelo de Colaboração 3C no Ambiente AulaNet, *Informática na Educação: Teoria e Prática*, v 7, n. 1, pp. 25-48, Porto Alegre, UFRGS. 1981.

[6] Huang, P. C. K. A Primitive Study of Logrolling in e-Negotiation. In: *Proceedings of the 36th Hawaii Conference on System Sciences* (HICSS´03). Hawaii. 2003.

[7] Neale, M. A. and bazerman, M. H. *Cognition and Rationality in Negotiation*. New York: The Free Press, 1991.

[8] Paula, M.M.V. *NegoSys: um ambiente de gestão do conhecimento na negociação*. Tese de Doutorado. Programa de Engenharia de Sistemas. COPPE. Universidade Federal do Rio de Janeiro. 2006.

[9] Raiffa, H. *The Art and Science of Negotiation*. Cambridge, MA, Harvard University Press, 1982.

[10] Susskind, L., Cruikshank, J.. *Breaking the Impasse: Consensual Approaches to Resolving Public Disputes*. New York: Basic Book. 1987.

Alterando o Grau de Colaboração de uma Ferramenta de Compartilhamento de Desktop por Meio de Plugins

Eduardo Telles Carlos
Tecgraf / DI / PUC-Rio
R. M. S. Vicente, 225
22453-900 – Rio de Janeiro, RJ
+55 21 2512-5984 - Brasil
etc@tecgraf.puc-rio.br

Alberto Raposo
Tecgraf / DI / PUC-Rio
R. M. S. Vicente, 225
22453-900 – Rio de Janeiro, RJ
+55 21 2512-5984 - Brasil
abraposo@tecgraf.puc-rio.br

RESUMO
Este artigo apresenta a experiência de desenvolvimento do VDTool (*Virtual Desktop Tool*), uma ferramenta para auxiliar processos colaborativos entre dois participantes, permitindo a troca de informações via captura e envio de tela. A aplicação surgiu da necessidade de simplificação de uma ferramenta de videoconferência existente. A partir de uma versão inicial, foram desenvolvidos *plugins* para incluir novos recursos de cooperação e coordenação na aplicação. O modelo de colaboração 3C serviu como mecanismo de análise para este desenvolvimento.

ABSTRACT
This paper presents the experience of developing VDTool (Virtual Desktop Tool), a tool to support collaborative processes between two participants, providing information exchange by means of screen capturing and transmitting. The application appeared as a consequence of the necessity of simplifying an existing videoconferencing tool. On top of an initial version, plugins were developed in order to add new cooperation and coordination facilities to the application. The 3C collaboration model was used as an analysis tool in this development.

Palavras-Chave
Colaboração, Modelo de Colaboração 3C, Plugins.

Keywords
Collaboration, 3C Collaboration Model, Plugins.

1. INTRODUÇÃO
Atualmente, com o crescente aumento da capacidade de processamento dos equipamentos e do desempenho da comunicação pelas redes, está se tornando cada vez mais viável, especialmente nas grandes companhias, o uso de ferramentas colaborativas. Tais ferramentas têm sido usadas para os mais variados propósitos, tais como realização de reuniões à distância e realização de trabalhos entre especialistas remotamente localizados.

Atendendo a essas necessidades, este trabalho iniciou-se com o desenvolvimento do CSVTool (*Collaboration Supported by Video Tool*), uma ferramenta para colaboração baseada em vídeo, projetada para prover recursos de colaboração para aplicações com nenhum ou pouco suporte à colaboração [1]. O projeto dessa ferramenta foi guiado pelas necessidades de uma grande companhia brasileira, tais como independência de plataforma, simplicidade de uso e suporte a múltiplos usuários sobre rede sem suporte a multicast.

O CSVTool, assim como qualquer outro sistema colaborativo, pode ser analisado de acordo com o modelo 3C de colaboração [2]. Este modelo, inicialmente proposto por Ellis [3], entende que o suporte computacional à colaboração pode ser realizado por meio da interação entre mecanismos de comunicação, coordenação e cooperação. Comunicação está relacionada à troca de mensagens e informações entre as pessoas. Coordenação diz respeito ao gerenciamento de pessoas, suas atividades e recursos. Cooperação, por sua vez, se relaciona à produção que ocorre em um espaço de trabalho compartilhado.

Apesar da separação dos 3Cs para fins de análise, comunicação, coordenação e cooperação não podem ser vistas de forma isolada, uma vez que existe uma constante interação entre elas [4]. Essa relação entre os 3Cs do modelo pode ser usada para guiar o projeto de uma ferramenta colaborativa qualquer, mesmo que ela esteja mais voltada para um dos 3Cs. Uma ferramenta de *chat*, por exemplo, que é uma ferramenta de comunicação, requer comunicação (troca de mensagens), coordenação (políticas de acesso) e cooperação (área de compartilhamento e registro das sessões).

Da mesma forma, o CSVTool, que também é mais voltado para a comunicação, possui não apenas recursos de comunicação (troca de informações por áudio, vídeo e texto), mas também de coordenação (controle de sessão, controles individualizados da recepção e envio de áudio/vídeo para cada participante) e cooperação (transmissão da área de trabalho – *desktop*).

O principal objetivo deste artigo é mostrar como a inclusão de novos "Cs" em uma aplicação colaborativa pode alterar seu grau de colaboração. Para isso, será apresentado um caso de estudo com o VDTool (*Virtual Desktop Tool*), uma ferramenta para auxiliar processos colaborativos entre dois participantes, permitindo a troca de informações pelo envio do *desktop* (via

SBSC Simpósio Brasileiro de Sistemas Colaborativos
20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 140-149.

captura de tela) e por uma ferramenta de bate-papo. Foram desenvolvidos *plugins* que implementam serviços de alguns "Cs", aumentando o grau de colaboração oferecido pela ferramenta.

Um segundo objetivo do artigo é mostrar que, seguindo o modelo 3C, o caminho nem sempre é unidirecional, i.e., nem sempre o que ser quer é acrescentar novos "Cs". Em alguns casos, as necessidades levam à retirada de "Cs" da ferramenta. Este caso é ilustrado na próxima seção, onde é explicado como surgiu o VDTool, que é uma versão simplificada do CSVTool. A Seção 3 apresenta trabalhos relacionados e a Seção 4 retoma a direção da inclusão de novos "Cs" no VDTool, apresentando os *plugins* desenvolvidos. A Seção 5 apresenta as conclusões e trabalhos futuros.

2. DO CSVTOOL AO VDTOOL

O CSVTool é uma ferramenta de videoconferência baseada no modelo de cliente/servidor, onde o servidor é responsável pelo gerenciamento dos clientes presentes em uma determinada sessão de colaboração. Os clientes, por sua vez, comunicam entre si através de áudio, vídeo e texto (via mensagens instantâneas). Além desses elementos de comunicação, a ferramenta também provê a funcionalidade de envio de *desktop* para visualização. Toda a interação é configurável, i.e., o participante tem controle sobre a comunicação, podendo escolher entre o envio/recebimento de áudio ou vídeo, ou ambos ao mesmo tempo.

Mesmo sendo uma ferramenta prioritariamente voltada para comunicação, o CSVTool, apresenta elementos de coordenação e cooperação enquadrando-se assim no modelo 3C [2]. A figura 1 mostra a decomposição das funcionalidades do CSVTool de acordo com o modelo 3C.

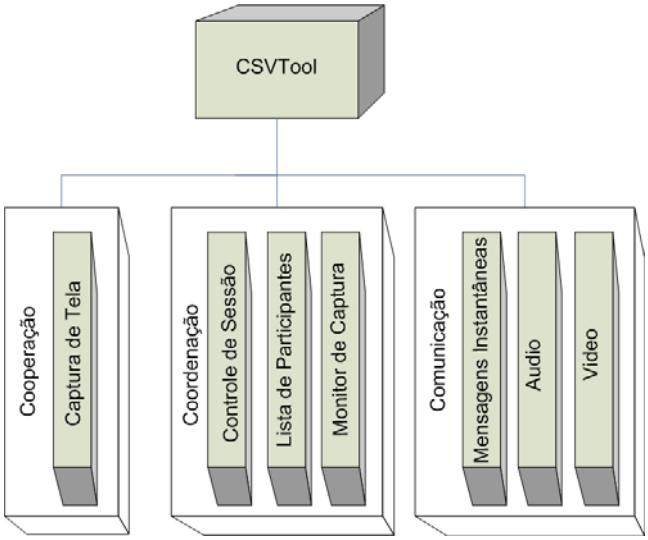


Figura 1- CSVTool e o modelo 3C

A separação das funcionalidades da ferramenta para análise trouxe uma nova visão para a solução de problemas de usabilidade e de implementação, mesmo tendo em mente que os "Cs" têm que interagir entre si, cada qual dando suporte para outro com a finalidade de aumentar o grau de colaboração.

O CSVTool vem sendo usado na empresa que financiou seu desenvolvimento. O processo de adoção de sistemas colaborativos em organizações é um processo muito delicado. Se o software for vendido da maneira usual, "de prateleira", ele pode não ser bem aceito na organização [5]. Na verdade, o processo de adoção de groupware em organizações é um processo dual, tanto de adaptação da organização de trabalho às condições da ferramenta, quanto de adaptação da ferramenta à organização do trabalho na companhia [6, 7]. Apesar de não discordarmos desta afirmação, acreditamos que as chances de sucesso na adoção do software serão maiores se o sistema colaborativo se adaptar completamente à real organização do trabalho na companhia. Por esta razão, o desenvolvimento do CSVTool tem apresentado uma natureza bastante dinâmica e até certo ponto imprevisível, sendo guiado pelas necessidades demandadas pelo seu uso no "mundo real".

O surgimento do VDTool é um exemplo da natureza dinâmica do processo de desenvolvimento do CSVTool. Com o uso do CSVTool na empresa, o recurso de envio de tela despertou o interesse de seus usuários para suprir um antigo problema. Em muitas situações, tais como treinamento e ajuda online, especialistas precisam demonstrar, de maneira mais clara, o que acontece em seus *desktops*, sem a necessidade de um encontro presencial. Nesses casos, a transmissão de *desktop* é um recurso mais importante que a transmissão de áudio e vídeo, visto que os especialistas geralmente se falam ao telefone durante essas situações. Esses usuários viram no CSVTool uma possibilidade para terem seus problemas resolvidos. Porém, alegaram que não queriam complexidades do sistema de videoconferência, tais como transmissão de áudio e vídeo (queriam continuar falando ao telefone) e necessidade de servidor (a colaboração é sempre entre duas pessoas apenas).

Como este não era o foco do CSVTool, houve a necessidade de criação de uma nova ferramenta com algumas características da original, porém menos complexa e que resolvesse o problema em questão.

Iniciou-se assim o desenvolvimento do VDTool, como uma versão simplificada do CSVTool, sem suporte a áudio, vídeo e sem servidor. Atualmente, a ferramenta VDTool auxilia processos colaborativos entre dois participantes, permitindo a troca de informações pelo envio do *desktop* (via captura de tela) e por uma ferramenta de bate-papo. Voltando à figura 1, o VDTool pode ser analisado à luz do modelo 3C como sendo apenas o serviço de cooperação do CSVTool (captura e envio de tela), acrescido de um serviço de coordenação (monitor de captura) e um serviço de comunicação (mensagem instantânea). Os demais serviços de comunicação foram eliminados (áudio e vídeo), bem como os demais serviços de coordenação (controle de sessão e lista de participantes, já que a comunicação no VDTool é um-para-um).

Na figura 2, de cima para baixo e da esquerda para direita, temos a tela remota recebida pelo usuário, a informação da área que está sendo enviada, a janela de eventos executados e, por fim, a troca de mensagens textuais do VDTool. Nesse caso, o usuário está enviando seu *desktop* e também recebendo o *desktop* remoto, de modo que existe tanto a tela remota recebida quanto o monitor de captura, onde ele vê a área da sua tela que está sendo enviada.

No momento, o VDTool tem sido usado com sucesso pelo grupo de usuários que solicitou seu desenvolvimento. Uma das suas características principais é ser multi-plataforma, atualmente disponível nas plataformas Windows, Linux, Solaris e Irix. Porém, a rigor, o VDTool tem poucos recursos de colaboração

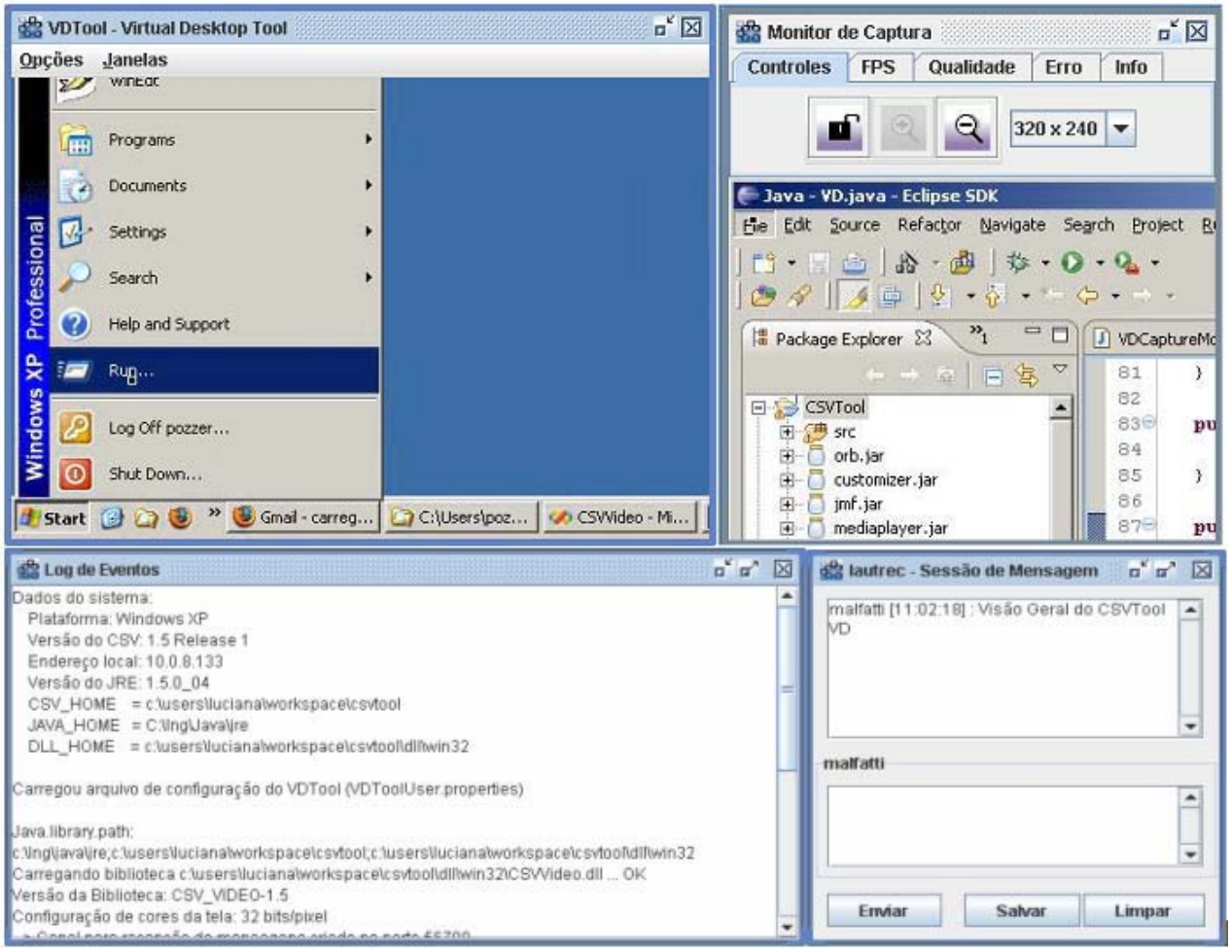


Figura 2 – Visão geral do VDTool

(i.e., poucos "Cs"). Com o intuito de aumentar o grau de colaboração do VDTool, foram inseridos *plugins* colaborativos na ferramenta.

A adição dos *plugins* trouxe novas funcionalidades para o sistema, acrescentando novos "Cs" à ferramenta e transformando-a novamente em uma ferramenta com bom grau de colaboração, embora com propósito totalmente diferente do CSVTool original. A figura 3 mostra a decomposição das funcionalidades do VDTool em relação ao modelo 3C. Nessa figura, os serviços que aparecem com fundo branco foram herdados do CSVTool, enquanto os que aparecem com o fundo colorido foram implementados pelos *plugins*.

A próxima seção esclarece a idéia dos *plugins* colaborativos como parte integrante de ferramentas colaborativas.

3. TRABALHOS RELACIONADOS

Uma vez que o VDTool já atendia os requisitos solicitados, o próximo passo foi voltar a acrescentar novos "Cs", tornando-a uma ferramenta com alto grau de colaboração, embora com propósito diferente do CSVTool.

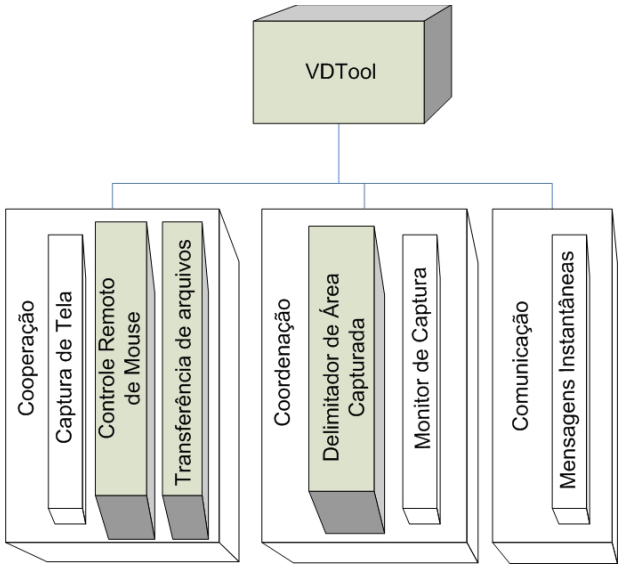


Figura 3-VDTool e o modelo 3C

Experiências similares, seguindo o modelo 3C, foram realizadas em [8,9], onde foram adicionados novos serviços de coordenação e cooperação a ferramentas de comunicação do ambiente AulaNet. A diferença em termos de implementação é que nos casos acima, foram adicionados componentes de software integrados a um *framework* de componentes. No caso do VDTool, como a aplicação não foi desenvolvida com o paradigma de componentes, a solução adotada para o acréscimo de novas funcionalidades foi o desenvolvimento de *plugins*.

Um *plugin* é um programa que deve interagir com outro programa para prover alguma funcionalidade particular [10]. O software principal determina a forma pela qual os *plugins* são incorporados ao sistema e a maneira que irão trocar informações. Uma das grandes vantagens do uso de *plugins* como parte integrante de um sistema é a não necessidade de recompilação ou substituição de partes do sistema original, o que pode gerar problemas de versão. Na maioria das vezes, os sistemas que dão suporte a *plugins* disponibilizam uma API que possibilita a interação entre eles. Os *plugins,* por sua vez, são implementados na forma de bibliotecas dinâmicas. Atualmente, vários programas profissionais disponibilizam APIs para desenvolvedores estenderem as funcionalidades de seus sistemas, aumentando assim o tempo até que seus sistemas se tornem obsoletos.

Na fase anterior ao desenvolvimento dos *plugins*, foi feita uma análise de sistemas colaborativos similares ao VDTool, explorando seus pontos fortes e fracos. A partir desta análise, foi possível o desenvolvimento voltado para implementar no VDTool algumas características interessantes encontradas nos sistemas analisados e também para suprir algumas limitações encontradas nos mesmos.

O VRVS (Virtual Room Videoconferencing System) [11] foi desenvolvido no início de 1997 e inicialmente usado dentro da comunidade HENP (High Energy and Nuclear Physics). Recentemente, o serviço foi estendido para outros grupos acadêmicos e de pesquisa. O VRVS é um sistema orientado para a *web,* para videoconferência e trabalho colaborativo. O desenvolvimento do sistema inclui hoje milhares de *hosts* registrados rodando o software em mais de 60 países. O VRVS fornece várias funcionalidades colaborativas como: envio e recebimento de áudio e vídeo, compartilhamento de *desktop* e aplicações e *chat* em várias plataformas. A análise deste software nos mostrou a dependência de outros sistemas para o funcionamento, como o MS Internet Explorer e o WinVNC, o que se torna uma dificuldade a mais para o usuário. Além disso, a existência de videoconferência, o coloca mais próximo do CSVTool do que do VDTool, que precisa manter sua simplicidade.

O RealVNC [12] foi desenvolvido com o intuito de melhorar o design e a implementação do VNC (Virtual Network Computing) [13]. O VNC é um sistema de compartilhamento de *desktop* para controlar outros computadores remotamente. Além do RealVNC, existem outras variantes desse sistema, como o TightVNC e o UltraVNC, que também foram analisadas para este trabalho. Essas ferramentas têm maior grau de colaboração que o VDTool em seu estágio antes dos *plugins*, visto que ele ainda não permitia o acesso ao *desktop* remoto, mas apenas sua visualização. Por esta razão, algumas das funcionalidades dos "VNCs" foram analisadas para serem incorporadas ao VDTool na forma de *plugins*. Algumas deficiências foram encontradas na parte de usabilidade, como por exemplo, a ausência de mecanismos de coordenação para evitar "brigas" pelo mouse controlado local e remotamente, e a ausência de algoritmos de escala na imagem recebida, assuntos que serão discutidos na Seção 4.1.

Outra ferramenta relacionada é o Sun SharedShell [14], que permitia o compartilhamento de um terminal de comando Unix (*shell*) entre usuários remotamente localizados, acrescentando também recursos de *whiteboard* compartilhado e gravação da sessão colaborativa. Seu intuito era permitir que o responsável pelo suporte Unix visse e interagisse com o terminal de comando do usuário. Além de restrita a Unix, essa ferramenta parece ter sido descontinuada.

4. PLUGINS COLABORATIVOS

O desenvolvimento dos *plugins* para o VDTool foi baseado na idéia do modelo 3C, i.e., cada *plugin* dando o suporte necessário para cada um dos "Cs" acrescentado, com a finalidade de aumentar assim o grau de colaboração. Uma vez implantados, os *plugins* têm que manter a coesão entre todos os "Cs" e, quando incorporados a um determinado sistema colaborativo, não fugir ao foco principal pelo qual a ferramenta original foi criada.

Seguindo esta idéia, são apresentados a seguir os *plugins* desenvolvidos e incorporados na ferramenta VDTool.

4.1 Controle Remoto de Mouse

O VDTool, por ser uma ferramenta colaborativa, deve prover o máximo de interação entre os participantes. Antes do desenvolvimento do *plugin* em questão, esta ferramenta não possuía nenhum esquema de acesso remoto, ou seja, não era possível compartilhar a aplicação remotamente, apenas visualizá-la. Este foi um dos problemas encontrados que diminuía a interação entre os usuários.

A idéia do *plugin* de controle remoto de mouse é disponibilizar a funcionalidade de controlar uma máquina remota através do mouse no mesmo ambiente do sistema colaborativo que incorporará este *plugin*. A finalidade principal do *plugin* é poder ampliar a capacidade do espaço de trabalho compartilhado, sendo portanto um *plugin* de cooperação.

A figura 4 mostra os processos envolvidos desde o início da execução do VDTool até a chamada do *plugin* de controle remoto de mouse.

Assim como a maioria dos *plugins* de outros sistemas, este foi implementado seguindo a estrutura de biblioteca dinâmica em Linguagem C. Como o VDTool foi desenvolvido na Linguagem Java, a incorporação deste utilizou a interface de comunicação JNI (Java Native Interface) [15]. Com isso, ocorre uma constante troca de informações entre o *plugin* e o VDTool.

Para o desenvolvimento deste *plugin*, houve o cuidado de se manter a característica de ser multi-plataforma, uma exigência da empresa que solicitou o desenvolvimento. Para tal, o código possui desvios condicionais destinados a cada plataforma.

A primeira informação a ser obtida para o funcionamento do *plugin* de controle remoto é a determinação da posição do mouse no sistema, para que ela possa ser enviada e posteriormente reconhecida na máquina remota (i.e., a máquina que está tendo seu *desktop* controlado). Neste ponto, definimos que uma posição é válida para ser enviada se o mouse estiver dentro da janela da

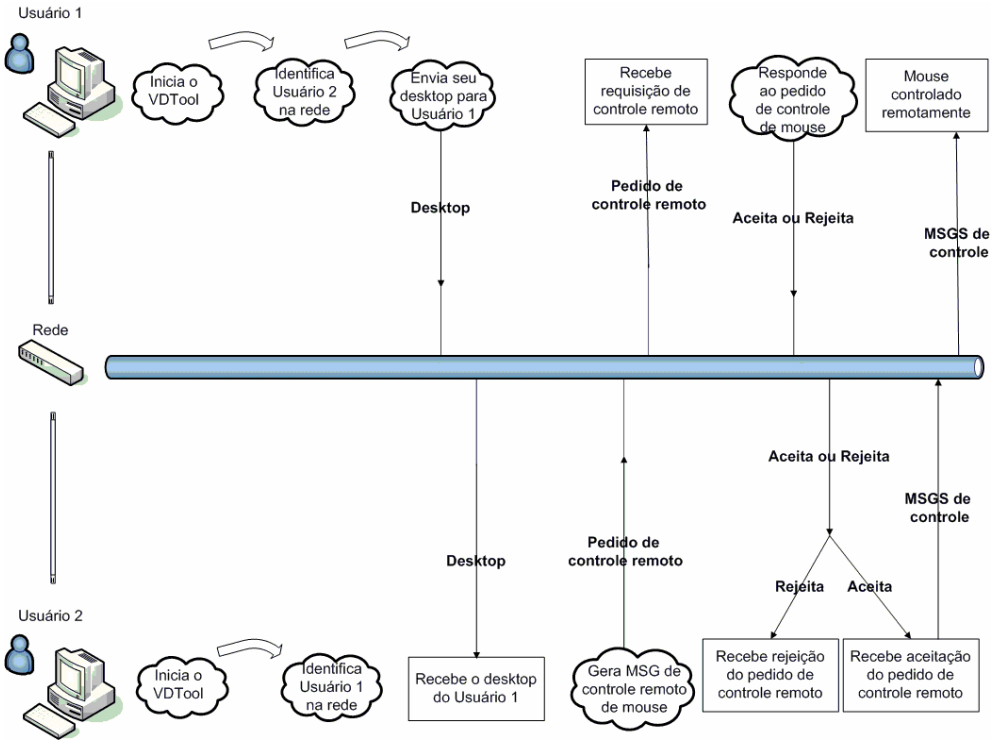


Figura 4-Diagrama do processo de acesso remoto

aplicação e sua coordenada seja diferente da última capturada, como mostrado na figura 5. Analogamente à posição do mouse, capturamos também os eventos referentes ao clique e *drag* do mouse para serem enviados caso a posição seja válida. Essas informações são então transmitidas via rede para a máquina remota.

A posição do mouse que chega à máquina remota não representa a posição real, em virtude da escala que a imagem sofreu na máquina local (figura 5), onde o usuário recebe toda a área de trabalho da máquina remota em uma imagem escalada (a escolha de um algoritmo ótimo para escalar imagens será explorado na Seção 4.1.1). Com isso a posição deve ser corrigida seguindo a seguinte fórmula:

$$Posição_real = \left(\frac{posição_na_imagem_escalada * dimensão_do_desktop}{zoom} \right)$$

A partir do momento que temos a posição real do mouse (calculada pela fórmula acima, a partir da posição enviada através da rede pelo usuário local) na máquina remota e um possível evento, podemos simular esta ação utilizando funções nativas de cada plataforma.

Neste ponto, o mouse está recebendo informações tanto do usuário remoto quanto do usuário local, podendo ocorrer uma "briga" pelo mesmo se os dois mandarem informações diferentes ao mesmo tempo. Para minimizar este problema, duas abordagens foram desenvolvidas. A primeira tenta explorar a idéia de *awareness* dando um *feedback* ao usuário remoto, informando que seu mouse está sendo controlado pelo outro usuário. Este *feedback* é dado pela alteração momentânea da aparência do ponteiro do mouse na máquina controlada (foi utilizado um cursor animado). Ao fim do controle remoto, as propriedades anteriores do ponteiro são restabelecidas.

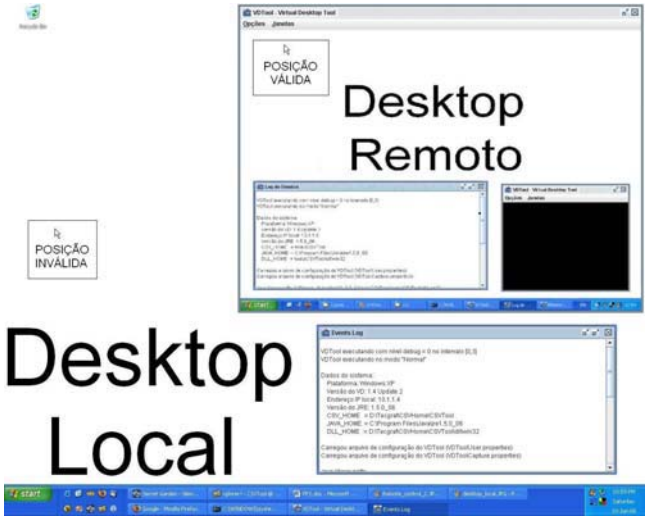


Figura 5-Posição válida do mouse (para ser enviada para a máquina remota)

Para tentar diminuir mais a possibilidade de "briga" pelo mouse, foi desenvolvida uma segunda abordagem que explora a idéia de um segundo ponteiro de mouse para o usuário remoto. A literatura não disponibiliza muito material para esta abordagem, contudo, o desenvolvimento do mesmo foi estudado e duas soluções surgiram: a primeira foi a simulação de um *driver* virtual USB que receberia eventos do usuário remoto e responderia às suas

ações [16] e a segunda, o desenho de um ponteiro "virtual" utilizando primitivas de desenho de cada plataforma e compartilhamento dos eventos de clique com o ponteiro real.

Neste trabalho apenas a segunda idéia foi implementada. Porém, ao fim do desenvolvimento, verificou-se que os resultados não foram satisfatórios, pois a operação de *redraw* das primitivas nativas de desenho do ponteiro não se mostrou eficiente para aplicações síncronas. Além disso, como os eventos de clique e *drag* do segundo ponteiro eram compartilhados com o ponteiro real, situações de conflito continuaram a ocorrer na geração desses eventos.

Por motivos de segurança, o cancelamento do controle remoto na máquina controlada é feito através da tecla *esc*, uma vez que poderia ocorrer "briga" pelo mouse neste momento.

4.1.1 Algoritmo de Escala

A necessidade do algoritmo de escala surgiu para solucionar dois problemas. O primeiro é que à medida que o usuário decide enviar uma parcela maior do seu *desktop* para o outro usuário, a área capturada que aparece no monitor de captura do VDTool teria que ser ampliada, aumentando assim o monitor de captura e o tráfego de dados na rede, o que não é desejado em algumas situações. O segundo problema é a necessidade de *scroll* na imagem recebida pela máquina que está realizando o controle remoto de mouse, problema que encontramos em ferramentas de acesso remoto, como os "VNCs" analisados.

Para a escolha do melhor algoritmo de escala de imagem para ferramentas síncronas foi levado em conta o que tivesse a melhor relação qualidade *vs*. desempenho. Foram estudados vários algoritmos presentes na literatura. Tal estudo identificou que o algoritmo dct, implementado em C, apresentou melhor qualidade [17]. Porém, é necessário levar em conta também o tempo de processamento do algoritmo. O algoritmo de escala deve ser rápido o suficiente para não se tornar o "gargalo" da aplicação e ter resultados satisfatórios. Com isso, os algoritmos disponíveis na Linguagem Java foram descartados, pois têm o tempo de processamento elevado em virtude da própria linguagem. O dct, apesar de sua melhor qualidade, também não obteve um desempenho muito bom. Ao final do estudo, chegou-se à conclusão que o algoritmo de *downsize rought* [18] obteve a melhor relação qualidade *vs.* desempenho, sendo por este motivo escolhido para ser incluído no *plugin* de controle remoto de mouse. A idéia do algoritmo de *downsize rought* é fazer uma média de cada componente RGB presente na imagem de acordo com um fator entre o tamanho da imagem original e o tamanho da imagem final como pode ser visto na figura 6. No caso mostrado na figura, a imagem original tem suas dimensões reduzidas à metade. Nesse caso, cada *pixel* da imagem reduzida é obtido como uma média de uma área de 2x2 *pixels* da imagem original.

4.2 Delimitador de Área Capturada

Para tornar mais claro ao usuário o que está sendo capturado e enviado da sua máquina para a outra, o VDTool utiliza o monitor de captura, ilustrado na figura 2. Quando ativo, além de dar o *feedback* necessário ao usuário, provê informações sobre o estado atual do sistema, como a banda passante e também disponibiliza opções para configuração do sistema, como a qualidade da imagem enviada. Porém, sua permanência na área de trabalho gera poluição visual, uma vez que o usuário que está enviando seu *desktop* perde área na tela, tendo assim que deslocar esta janela sempre que necessário para desempenhar outras atividades. Este foi mais um dos problemas identificados na ferramenta VDTool, que acabam por atrapalhar a colaboração entre os usuários.

	1	2	3	4
1	R = 255 G = 0 B = 0	R = 197 G = 208 B = 48	R = 13 G = 13 B = 45	R = 255 G = 16 B = 78
2	R = 153 G = 153 B = 255	R = 109 G = 94 B = 29	R = 0 G = 126 B = 244	R = 41 G = 0 B = 10
3	R = 77 G = 102 B = 0	R = 255 G = 255 B = 0	R = 10 G = 154 B = 247	R = 148 G = 199 B = 45
4	R = 33 G = 48 B = 204	R = 0 G = 0 B = 255	R = 47 G = 255 B = 255	R = 255 G = 255 B = 255

	1	2
1	R = 178 G = 114 B = 83	R = 77 G = 39 B = 94
2	R = 91 G = 101 B = 115	R = 115 G = 216 B = 200

Figura 6-Ilustração do algoritmo de escala *downsize rought*

Algumas soluções foram levantadas, como a diminuição da área ocupada pelo monitor de captura no *desktop*. Porém, isto ainda causaria certo incômodo e perderíamos o poder de *awareness* dos elementos da interface gráfica. Outra solução estudada foi fazer o uso de transparência em cima da janela do VDTool, porém isto iria impossibilitar o usuário de acessar possíveis elementos que estejam embaixo da janela. A melhor alternativa encontrada para tal problema foi o desenvolvimento de um *plugin* que delimita a área capturada através do desenho de linhas no próprio *framebuffer* do usuário que, ao mesmo tempo em que forneça o *feedback* ao usuário, permita que o mesmo minimize o monitor de captura quando necessário e continue trabalhando com toda a área de trabalho. Por ter o intuito de ajudar no gerenciamento da interação entre os usuários e ter funcionalidade complementar à do monitor de captura, este *plugin* é considerado um serviço de coordenação.

Assim como o *plugin* de controle remoto de *mouse*, este foi implementado em forma de biblioteca dinâmica na Linguagem C e com diretivas condicionais de compilação para cada plataforma. A figura 7 mostra a troca de informações entre o VDTool e o *plugin* de delimitação de área capturada.

Como o delimitador de área capturada foi implementado utilizando código nativo na Linguagem C, houve mais uma vez a necessidade da utilização da JNI para que fosse possível a comunicação entre o VDTool e o *plugin*. Nesse ponto houve alguns problemas relativos à plataforma Unix e à posição do ponteiro do mouse nas extremidades da tela. No *plugin* delimitador de área capturada, as rotinas de desenho no *framebuffer* eram chamadas e, posteriormente, o código em Java era chamado novamente. Este esquema foi válido para a plataforma Windows, porém inválido para as plataformas Unix, uma vez que nesta era impossível sair do *loop* de *redraw* das rotinas de desenho no *framebuffer* e retornar ao código Java. Para resolver este problema, foram criados executáveis para serem chamados a partir do código em Java e executados em paralelo com o código em Java do VDTool. Dessa forma, esses executáveis desempenham a mesma funcionalidade do *plugin* desenvolvido para a plataforma Windows.

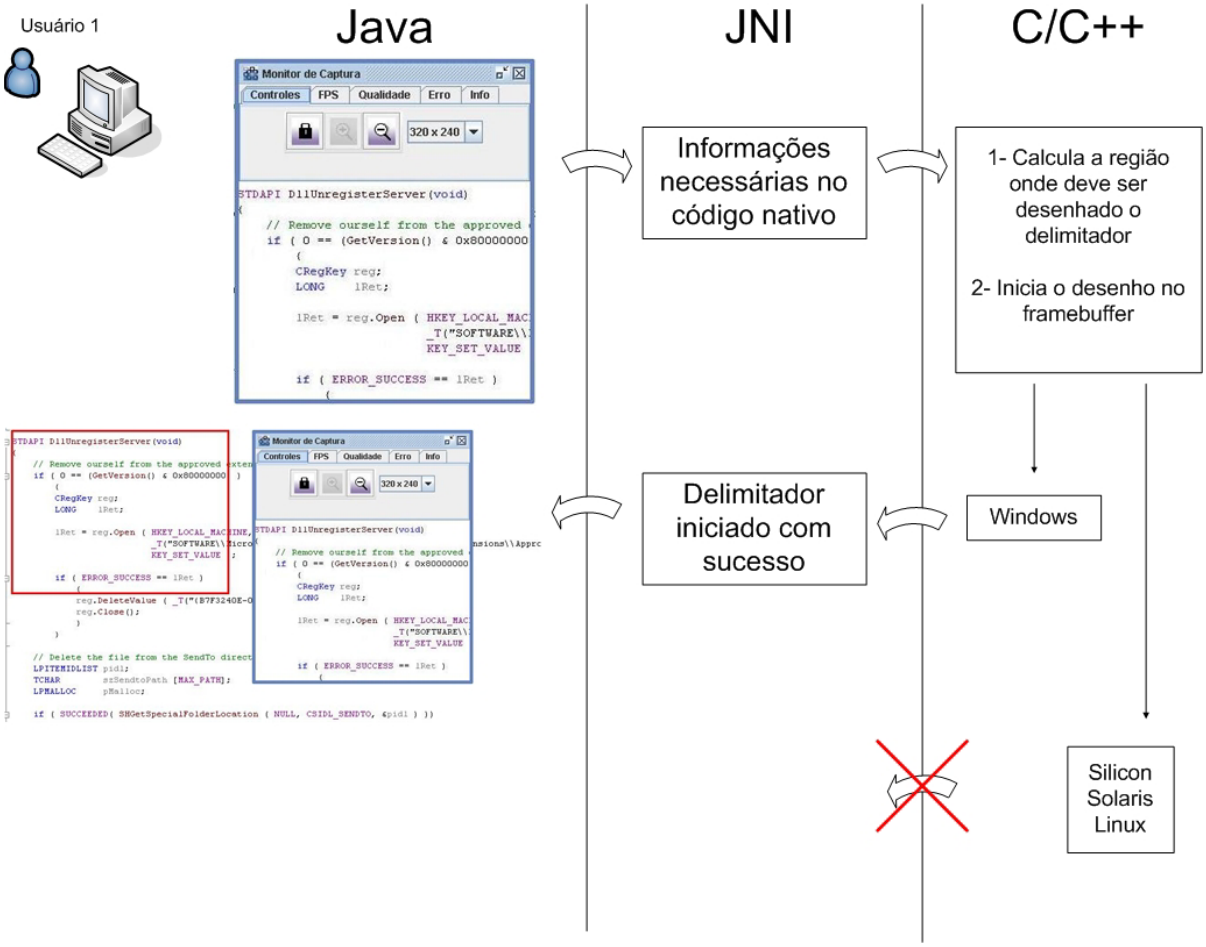


Figura 7 – Delimitador de área capturada e o VDTool

4.3 Transferência de Arquivos

A transferência de arquivos entre usuários é uma funcionalidade muito comum e útil em ferramentas colaborativas. A maior motivação para o desenvolvimento desse *plugin* foi a ausência desta funcionalidade no VDTool e a presença na maioria das ferramentas analisadas anteriormente. Porém, em algumas dessas ferramentas, a transferência de arquivos ocorre em apenas um sentido, i.e., sendo permitido apenas o envio de arquivos da máquina que está controlando a outra remotamente (*upload* de arquivos). O RealVNC, por exemplo, faz a transferência de arquivos bidirecional em sua versão comercial, mas não faz nenhum tipo de transferência em sua versão gratuita.

Assim como no caso do controle remoto do mouse, o *plugin* de transferência de arquivos amplia a capacidade do espaço de trabalho compartilhado, se enquadrando como um serviço cooperação no modelo 3C.

A maioria das aplicações colaborativas que disponibilizam a funcionalidade de transferência de arquivos utiliza a própria janela da aplicação para realizar essa transferência, i.e., a transferência é feita a partir de um elemento da interface do programa. Como um dos focos deste trabalho é o reuso de *plugins* colaborativos em diversas ferramentas da forma mais transparente possível, tanto para o desenvolvedor quanto para o usuário final, essa abordagem de usar a própria janela da aplicação não foi utilizada.

Para implementação deste *plugin* foram estudadas duas opções: a primeira fazendo a transferência de arquivos entre usuários utilizando *drag and drop* e a segunda fazendo a transferência através do *context menu* do Windows.

A idéia de transferência via *drag and drop* pode ser vista de duas formas. A primeira forma é para realizar o *upload* de arquivos ou pastas entre o sistema da máquina local (*drag*) para a janela do VDTool, representando a máquina remota (*drop*), como pode ser visto na figura 8. A segunda forma é para realizar o *download* de arquivos ou pastas da máquina remota através da janela do VDTool (*drag*) para a máquina local (*drop*).

Inicialmente, esta seria a única forma de transferência de arquivos implementada na forma de *plugins*, porém houve a necessidade de termos outra forma de transferência devido a problemas encontrados em testes na parte referente à execução de *download* de arquivos ou pastas da máquina remota para a máquina local. A primeira etapa para realizar a transferência de dados entre a máquina remota e a máquina local via *drag and drop* é a identificação da ação de *drag* de algum arquivo ou pasta, ou um conjunto deles. Identificado este evento, executado pelo usuário

que está comandando remotamente a outra máquina, a próxima etapa é disparar o evento de transferência dos elementos que estejam sofrendo *drag*. O início deste evento só será efetivado caso os elementos que estejam sendo "arrastados" passem por uma das bordas do *desktop* da máquina controlada ilustrado em vermelho na figura 9.

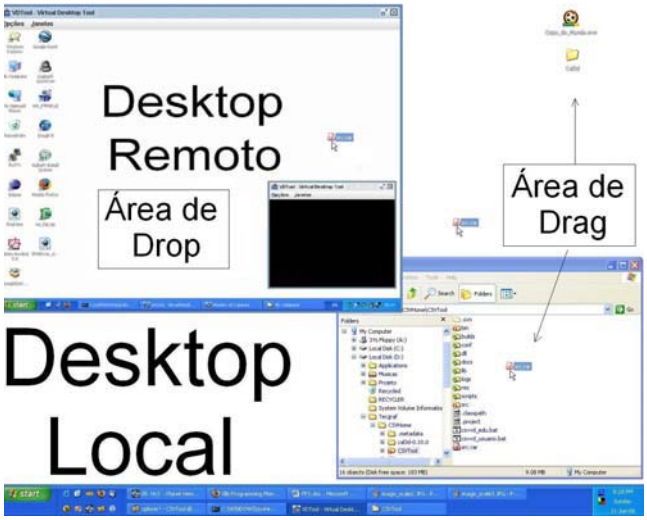


Figura 8-Áreas de *drag and drop* para *upload* de arquivos

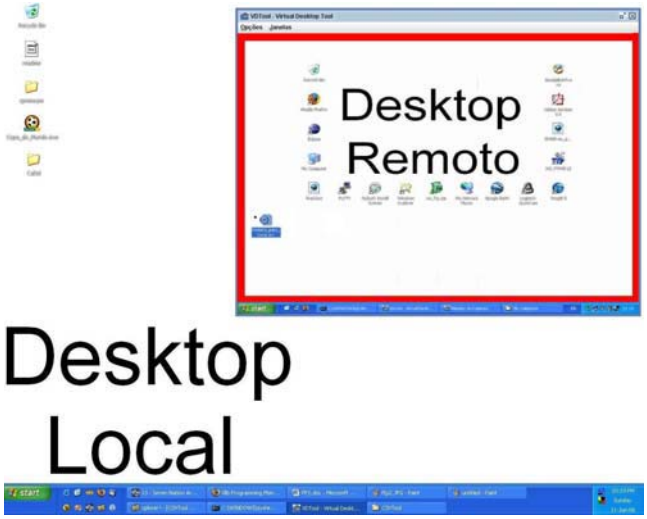


Figura 9-Área de inicio do *download* em vermelho

Ao entrar na área delimitada, o *plugin* identifica o caminho para todos os elementos que estão sofrendo *drag* para que posteriormente possa iniciar a transferência dos mesmos para a outra máquina. Neste ponto justifica-se a inclusão de uma maneira alternativa de executar esta funcionalidade, pois nem sempre é possível identificar o caminho para todos os elementos de maneira rápida (necessidade em ferramentas síncronas), uma vez que esta informação é requisitada ao sistema operacional da máquina remota e não é recebida imediatamente. Esta situação não ocorre no *upload*, pois o sistema operacional da máquina remota não precisa ser consultado.

A outra forma de transferência de arquivos implementada para este *plugin* foi a utilização do *context menu* do Windows (figura 10). Desta forma os problemas mencionados anteriormente são contornados, pois a transferência é iniciada imediatamente após a sua solicitação. Como esta parte do *plugin* envolve a inserção de novas entradas no registro do Windows, ao finalizar a conexão entre os participantes, o *plugin* deve ser chamado novamente para retirar a entrada no registro do sistema operacional.

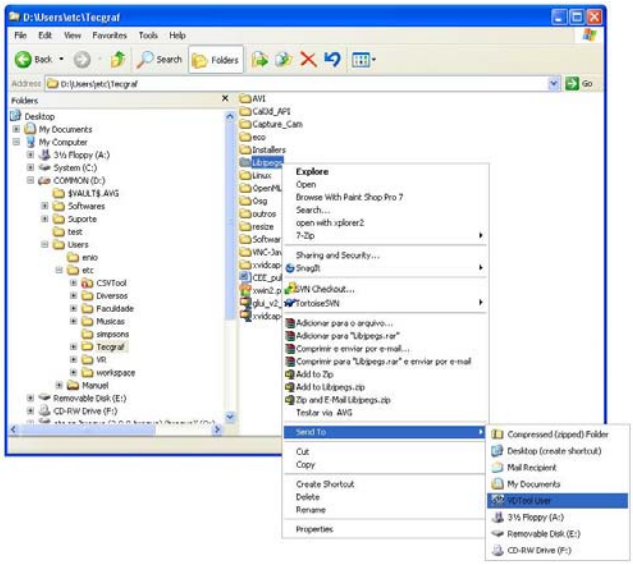


Figura 10-Transferência de arquivos via *context menu*

Diferentemente dos outros *plugins*, este só foi implementado para a plataforma Windows, por ser o sistema operacional mais utilizado e aparentemente o mais simples para o desenvolvimento, em uma primeira análise. Para manter o requisito de portabilidade multi-plaforma, este *plugin* ainda precisa ser implementado nas outras plataformas.

5. CONCLUSÕES

A integração visual entre aplicações executadas em locais distintos permite que profissionais geograficamente distribuídos trabalhem colaborativamente, reduzindo as barreiras de comunicação e aumentando a produtividade. Essa possibilidade tem atraído a atenção das empresas, que cada vez mais buscam sistemas colaborativos adequados às suas necessidades.

Porém, uma vez adotada nas organizações, a tecnologia só se torna efetiva se a ela for atribuída um significado e se ela realmente for utilizada [7]. O exemplo do CSVTool/VDTool mostra que este "significado", do ponto de vista dos usuários na empresa, pode ser surpreendente aos olhos dos desenvolvedores de groupware. Cabe aos desenvolvedores, portanto, buscar o entendimento deste significado e encontrar os meios tecnológicos para implementá-los em suas aplicações.

O modelo 3C, como ferramenta de "engenharia" de domínio, se mostrou eficiente para ajudar a cobrir a distância existente entre os requisitos sociais e a viabilidade tecnológica [19]. O caso do VDTool mostra como a análise, à luz do modelo 3C, revela que a exclusão e a inclusão de diferentes "Cs" podem alterar significativamente a funcionalidade de uma aplicação colaborativa, adequando-a a novos propósitos. Dessa forma, abre-

se caminho para que os sistemas colaborativos se adaptem às necessidades dos usuários, e não o contrário, o que aumenta as chances de sucesso do software.

A maneira utilizada para incluir novas funcionalidades colaborativas no VDTool foi pela implementação de *plugins* que realizem tais funcionalidades. Um dos objetivos do desenvolvimento via *plugins* é o possível reuso dos mesmos em outras aplicações colaborativas além do VDTool. Para isso, seria necessário que a outra aplicação tenha uma API flexível para interagir com os *plugins* desenvolvidos. Uma possibilidade alternativa seria o desenvolvimento baseado em componentes de software, como realizado em outras aplicações [2]. Entretanto, essa opção exigiria uma reimplementação do VDTool e por isso não foi escolhida.

Com relação às outras ferramentas analisadas, pode-se dizer que o VDTool é diferente de todas elas, embora não necessariamente "mais colaborativo". Com relação ao VNC, o VDTool permite a transferência de arquivos via *drag and drop* entre a máquina local e a remota, o que não acontece no VNC, onde o *desktop* da máquina local "desaparece" quando se acessa o *desktop* da máquina remota. Com relação ao Sun SharedShell, o VDTool compartilha todo o *desktop*, e não só uma janela de comandos. Porém, o VDTool não possui o recurso de gravação de sessão do Sun SharedShell. Quanto ao VRVS, que é um software mais abrangente, o VDTool é, por requisito de projeto, mais simples. Porém, uma possível integração do CSVTool com o VDTool poderá se aproximar do VRVS, em termos de recursos de colaboração. Entretanto, independente dos recursos de colaboração, uma inovação do VDTool é o uso de *plugins*, que permitem alterar seu grau de colaboração, inserindo novas funcionalidades.

Voltando a mencionar a dualidade no processo de adoção de uma tecnologia em uma organização, podemos dizer que a transformação do CSVTool no VDTool foi uma adaptação da ferramenta às pessoas. Por outro lado, a implementação dos *plugins* para aumentar o grau de colaboração do VDTool está sendo uma tentativa de oferecer mais recursos aos usuários, que poderão adaptar seus processos de trabalho a estes novos recursos.

Uma das próximas etapas do trabalho é verificar a aceitação desses novos recursos junto aos usuários, bem como realizar experimentos que comprovem a alteração (preferencialmente de forma positiva) no grau de colaboração provido pelos novos recursos do VDTool. Outra atividade futura será a avaliação da possibilidade de incluir os *plugins* do VDTool no CSVTool, tornando este último uma ferramenta de colaboração mais completa, com mais recursos de cooperação principalmente, como a transferência de arquivos e controle remoto de mouse.

6. AGRADECIMENTOS

Agradecemos aos Profs. Marcelo Gattass, coordenador do Tecgraf (Grupo de Tecnologia em Computação Gráfica) e Hugo Fuks, coordenador do Groupware@LES, DI, PUC-Rio, DI, PUC-Rio pelo constante apoio ao desenvolvimento deste trabalho. O Tecgraf é um grupo prioritariamente financiado pela Petrobras.

7. REFERÊNCIAS

[1] Pozzer, C. T., Lima, L. S., Raposo, A. B., Vieira, C. J. G., A Multi-user Videoconference-based Collaboration Tool: Design and Implementation Issues. In *Proceedings of the 9th International Conference on CSCW in Design (CSCWD 2005)* (Coventry, U.K.), 2005, 547-552.

[2] Fuks, H., Raposo, A. B., Gerosa, M.A., Lucena, C. J. P. Applying the 3C Model to Groupware Development. *International Journal of Cooperative Information Systems, 14, 2-3* (Jun.-Sep. 2005), 299-328.

[3] Ellis, C. A., Gibbs, S. J., Rein, G. L. Groupware: Some Issues and Experiences. *Communications of the ACM, 34, 1* (Jan. 1991), 38-58.

[4] Pimentel, M., Fuks, H., Lucena, C.J.P. Mediated Chat 2.0: Embedding Coordination into Chat Tools. In *Conference Supplement of the 6th International Conference on the Design of Cooperative Systems (COOP 2004)* (Amsterdam, The Nederlands), 2004, 99-103.

[5] Grudin, J. Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM, 37, 1* (Jan. 1994), 92-105.

[6] Bardram, J.E. Organizational Prototyping: Adopting CSCW Applications in Organisations, In G. Mark et al. (organizers), CSCW 96 Workshop: *Introducing groupware in organizations: What leads to successes and failures?* 1996.

[7] Orlikowski, W. J. The Duality of Technology: Rethinking the Concept of Technology in Organizations. *Organization Science, 3, 3* (Ago. 1992), 398-427.

[8] Fuks, H., Raposo, A.B, Gerosa, M.A., Pimentel, M., Lucena, C.J.P. Suporte à Coordenação e à Cooperação em uma Ferramenta de Comunicação Textual Assíncrona: Um Estudo de Caso no Ambiente Aulanet. In *Anais do I Workshop Brasileiro de Tecnologias para Colaboração (WCSCW 2004)* (Ribeirão Preto, Brasil), 2004, 173-180.

[9] Gerosa, M. A., Raposo, A.B., Fuks, H., Lucena, C.J.P. Uma Arquitetura para o Desenvolvimento de Ferramentas Colaborativas para o Ambiente de Aprendizagem AulaNet. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE 2004)* (Manaus, Brasil), 2004, 168-177.

[10] Plug-in. *Wikipedia*. Acesso em 25 de Julho, 2006, a partir do Answers.com Web site: http://www.answers.com/topic/plugin

[11] Virtual Rooms Videoconferencing System. Acesso em 26 de Julho de 2006: http://www.vrvs.org

[12] RealVNC. Acesso em 26 de Julho de 2006: http://www.realvnc.com

[13] Richardson, T., Stafford-Fraser, Q., Wood, K.R., Hopper, A. Virtual network computing. *IEEE Internet Computing, 2, 1* (Jan.-Fev. 1998), 33-38.

[14] Yankelovich, N., "Bo" Begole, J., Tang, J. C. Sun SharedShell tool. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW 2000)* (Philadelphia, USA), 2000, 351.

[15] Sun Microsystems, Inc. *JNI (Java Native Interface) Specification*. Acesso em 25 de Julho, 2006: http://java.sun.com/j2se/1.4.2/docs/guide/jni/spec/jniTOC.html

[16] Wierzchowski, G. *Second Mouse in X mini-HOWTO*. Acesso em 26 de Julho de 2006: http://www.faqs.org/docs/Linux-mini/XFree86-Second-Mouse.html

[17] Martucci, S.A. Image resizing in the discrete cosine transform domain. In *Proceedings of the International Conference on Image Processing (ICIP'95), Volume 2*, 1995, 2244-2247.

[18] Lacroix, D. *Resize Images*. Acesso em 26 de Julho de 2006: http://lab.erasme.org/resize_image/index.html

[19] Ackerman, M. S. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. In *Human-Computer Interaction 15, 2&3* (2000), 179-203.

Desenvolvimento de Aplicações de Democracia Eletrônica em Larga Escala Utilizando Grades Computacionais

Rogério Augusto Rondini
Programa de Mestrado em Informática
Univ. Católica de Santos (UNISANTOS)
R. Dr. Carvalho de Mendonça, 144 - 11070-906 - Santos - SP
rarondini@paradygma.com.br

Hermes Senger
Programa de Mestrado em Informática
Univ. Católica de Santos (UNISANTOS)
R. Dr. Carvalho de Mendonça, 144 - 11070-906 - Santos - SP
senger@unisantos.br

Cléver R. G. de Farias
Depto. de Física e Matemática
Univ. de São Paulo (FFCLRP/USP)
Av. Bandeirantes, 3900 - 14040-901 - Ribeirão Preto - SP
farias@ffclrp.usp.br

Resumo

Democracia eletrônica pode ser definida como o uso de Tecnologias de Informação e Comunicação para intensificar a participação ativa dos cidadãos e dar suporte à colaboração entre governos e sociedade civil na elaboração de políticas públicas. Para que ocorra interação entre sociedade civil e governantes, é necessário a criação de ambientes colaborativos para acesso em larga escala. Tais ambientes podem ser vistos como um local onde discussões de interesse público ocorrem e onde o conteúdo dessas discussões possam ser encontrados e acessados por qualquer pessoa em qualquer lugar. A evolução das plataformas de grades computacionais e das tecnologias de desenvolvimento baseado em serviços permite a criação de ambientes colaborativos através do compartilhamento, coordenação e gerenciamento de recursos de diversos tipos. Nesse sentido, este artigo apresenta a arquitetura e implementação de um conjunto de serviços, baseados em grades computacionais, para o gerenciamento, busca e recuperação de informações. Tais serviços são utilizados na construção de uma ferramenta de democracia eletrônica para dar suporte à colaboração em larga escala.

Abstract

Electronic democracy (e-democracy) can be defined as the use of Information and Communication Technology to improve the participation of citizens in the political process, as well as to support the collaboration between the different instances of government and community for policy-making purposes. In order to facilitate the interaction between government and citizens, the creation of a wide scale collaborative environment is necessary. Such environment can be seen as a place where public discussions take place and published contents can be found and accessed by everyone and everywhere. The evolution of grid platforms and service-oriented technologies allows the creation of a collaborative environment through the management, coordination and sharing of different types of computing resources. This paper presents the architecture and implementation of a basic set of grid-based services for information management, search and retrieval. These services are used in the development of e-Democracy application to support wide scale collaboration.

Palavras-chave

Democracia eletrônica, governo eletrônico, ambientes colaborativos, grades computacionais, colaboração, larga escala.

Keywords

E-Democracy, e-government, collaborative environment, grid computing, wide scale, collaboration.

1. INTRODUÇÃO

A disseminação dos serviços eletrônicos teve um grande impulso com o surgimento e a legalização de identidades digitais e assinaturas eletrônicas, dentre outras técnicas de segurança. Não apenas empresas, mas também governos estão investindo fortemente no uso de Tecnologias de Informação e Comunicação (TIC) como forma de oferecer serviços aos cidadãos. No Brasil existe um número crescente de portais com serviços oferecidos tanto pelo governo federal quanto pelos governos estaduais e municipais. Através desses serviços, cidadãos podem pagar impostos, obter licenças, permissões, dentre outros. Nesse sentido, espera-se também que as TIC melhorem o relacionamento entre cidadãos (ou seus grupos de interesse), administrações, políticos e governos.

Democracia eletrônica pode ser definida como o uso de TIC e Comunicação Mediada por Computador (CMC) para in-

SBSC Simpósio Brasileiro de Sistemas Colaborativos

20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 150-160.

tensificar a participação ativa dos cidadãos e dar suporte à colaboração entre os diversos atores, tais como cidadãos, governos, Organizações Não Governamentais (ONGs), etc., na elaboração de políticas públicas [15]. Em função das facilidades advindas do uso das TIC, cidadãos, empresas e outras organizações têm expectativas crescentes acerca da democracia eletrônica em geral. Dentro de poucos anos, espera-se que várias fases do chamado ciclo político [15] venham a ser apoiadas por uma plataforma integrada, colaborativa e eficiente, disponibilizada através da Internet. Nesse sentido, este artigo propõe a adoção de plataformas de grades computacionais como infra-estrutura de desenvolvimento e execução de aplicações de democracia eletrônica.

Inicialmente, as tecnologias de grade computacional tiveram como proposta o compartilhamento de recursos computacionais para o processamento de aplicações paralelas de alto desempenho. Porém, tais tecnologias evoluiram nos últimos anos com intuito de promover o compartilhamento, coordenação e gerenciamento de recursos dos mais variados tipos, permitindo, assim, sua utilização no desenvolvimento de aplicações não científicas [26], tais como comércio eletrônico, governo eletrônico, telecomunicações e aplicações comerciais em geral.

Grades computacionais permitem criar ambientes colaborativos através de coleções de serviços e abstrações capazes de integrar os recursos em diferentes níveis, desde recursos computacionais, tais como dados e ferramentas de software, até recursos representando conhecimento [25]. Nesse sentido, plataformas de grades computacionais podem ser utilizadas para superar muitos dos desafios impostos pelas aplicações de democracia eletrônica. Por exemplo, atualmente nenhuma infra-estrutura de TIC escala adequadamente para prover níveis adequados de colaboração entre milhões de cidadãos e governantes.

Adicionalmente, a disponibilização de informações a qualquer momento e em qualquer lugar é uma precondição para a participação dos cidadãos nesse processo. Nesse sentido, grades computacionais podem ser utilizadas para acessar e integrar recursos inerentemente distribuídos e heterogêneos, tais como relatórios, bases de dados das agências governamentais, fóruns de discussões e aplicações legadas, nos mais diversos níveis da esfera governamental, como esperado em grandes aplicações de democracia eletrônica. Nesse tipo de cenário, é imprescindível a interação entre cidadãos e administrações nos níveis municipal, estadual e federal.

Este artigo apresenta um conjunto básico de serviços baseados em grades computacionais para o desenvolvimento de aplicações de democracia eletrônica. A proposta aqui apresentada envolve a identificação dos requisitos para o desenvolvimento de aplicações de democracia eletrônica, definição da arquitetura, protótipo e implementação de um conjunto de serviços que atendam aos requisitos identificados.

O artigo está organizado da seguinte forma: a seção 2 apresenta uma breve introdução sobre democracia eletrônica e seus desafios; a seção 3 discute os principais conceitos e a evolução da tecnologia de grades computacionais; a seção 4 discute a utilização de grades computacionais como mecanismo de suporte à colaboração em larga escala e apresenta a arquitetura da ferramenta; a seção 5 apresenta alguns trabalhos relacionados; a seção 6 faz uma breve análise do uso de ferramentas colaborativas em aplicações de democracia eletrônica; finalmente, a seção 7 apresenta algumas considerações finais e perspectivas de trabalhos futuros.

2. DEMOCRACIA ELETRÔNICA

A adoção de TIC como forma de intensificar a participação dos cidadãos no processo democrático enfrenta diversos desafios, tecnológicos e também sócio-econômicos. Nesse sentido, ferramentas de TIC terão o papel de auxiliar no rompimento das barreiras organizacionais e culturais inerentes de cada país que venham adotá-las, além de permitirem a colaboração dos atores envolvidos no processo político.

A democracia eletrônica pode ser alcançada através de diversas formas de interação colaborativa, tais como: campanhas eleitorais, pesquisas de opinião, comunicação entre os representantes eleitos e a população em geral, disponibilidade on-line da legislação e processos legislativos que promovam a participação dos cidadãos, ao invés da simples disseminação da informação.

O relacionamento entre governo e sociedade pode ser classificado de acordo com a natureza e direção das interações estabelecidas durante o chamado ciclo político. Três tipos de relacionamentos são freqüentemente encontrados [18, 20, 25]:

- *Informação.* Relação de sentido único em que o governo fornece informação para uso dos cidadãos;
- *Consulta.* Relação de sentido duplo entre cidadãos e governo onde os cidadãos, após consulta, compartilham suas idéias e opiniões;
- *Participação ativa.* Relação de sentido duplo baseada em parcerias com o governo, onde os cidadãos participam ativamente na formulação das políticas, por exemplo, definindo agendas e propondo opções de políticas.

A entrega e disseminação da informação são normalmente alcançadas através do uso de websites e portais do governo. Contudo, a maior parte dos portais falha ao prover um suporte inadequado a usuários não especializados na busca por determinada informação ou serviço. Facilidades tais como mecanismos de busca e esclarecimento de perguntas freqüentes devem ser providos de modo a facilitar a busca da informação e a localização dos serviços.

A consulta aos cidadãos pode ser implementada de diferentes formas. Os websites e portais do governo normalmente provêem endereços eletrônicos que podem ser utilizados para o envio de comentários e sugestões. Formulários on-line também podem ser utilizados para coletar informações dos usuários dos portais. Listas de endereços também podem ser utilizadas para a circulação de documentos entre possíveis interessados e a posterior coleta de comentários. Fóruns de discussão on-line, chats e grupos de notícias também podem ser utilizados para consulta. Essas ferramentas provêem uma forma conveniente para a coleta e a entrega aos representantes eleitos de propostas, geradas pelos cidadãos ou por organizações representativas. Pesquisas de opinião on-line também podem ser utilizadas na coleta da opinião dos cidadãos em tópicos específicos.

Finalmente, a participação ativa pressupõe níveis mais elevados de envolvimento da população e apresenta maior potencial de influenciar nas decisões políticas. O suporte ferramental para esse tipo de interação inclui não apenas fóruns de discussão on-line, mas também referendos on-line e requisições eletrônicas ao poder executivo e legislativo.

Em geral, os fatores críticos para o desenvolvimento de ferramentas de democracia eletrônica, e subsequente adoção por parte dos cidadãos, incluem [22]: usabilidade, acessibilidade e segurança. Nesse sentido, a implementação dessas ferramentas deve ter como premissas: (i) a possibilidade de o usuário optar por acesso ao sistema como anônimo ou autenticado, nesse último caso, garantindo segurança das informações pessoais; (ii) balancear a definição de padrões com a adoção de interfaces genéricas, com o objetivo de atender a públicos variados; (iii) navegação simplificada através das informações; e (iv) ser um ambiente seguro e confiável.

Além desses requisitos básicos, a OECD (*Organisation for Economic Co-Operation and Development*) identificou outros cinco desafios fundamentais para o sucesso das ferramentas de democracia eletrônica [20, 21, 22]:

1. *Problema de Escala* - Um dos principais desafios para implementação de democracia eletrônica é a transformação de discussões de âmbito local em opiniões que possam atingir toda uma nação. Em uma sociedade de massas, é muito comum que opiniões se percam dentre as milhões de possíveis opiniões sobre um mesmo assunto. Nesse sentido, é necessário que ferramentas de TIC atendam tanto a demanda do cidadão quanto dos governantes, de modo que as opiniões de cidadãos sejam compartilhadas entre os cidadãos que participam do processo político, e, por outro lado, governantes tenham acesso a todas as opiniões e possam responder a cada uma delas, sem privilegiar indivíduos ou grupos.

2. *Capacitação e Construção da Cidadania* - A capacitação e a construção da cidadania pode se dar de três maneiras: (i) por um processo de educação formal sobre a constituição, as leis e as práticas políticas de cada país; (ii) incentivo à participação no debate das questões locais; ou (iii), pela participação ativa da sociedade no processo político. Nesse sentido, a utilização de ferramentas de TIC tem tanto o desafio, como a oportunidade de contribuir para uma sociedade participativa.

3. *Garantia de Coerência das Informações* - Garantir a coerência das informações está diretamente relacionado a conceitos de *Gerência de Conhecimento*. Durante o processo de formulação de políticas públicas são gerados centenas ou milhares de documentos e opiniões que devem ser mantidos durante todo o ciclo. Todo o conhecimento produzido durante o processo deve estar disponível para consulta dos atores participantes, de forma que possam ser analisados, alterados e respondidos eletronicamente. Para direcionar esse desafio pode-se utilizar técnicas de trabalho e argumentação colaborativa através do uso de computadores. A criação de vocabulários comuns é muito importante para facilitar o desenvolvimento de ferramentas integradas. Alguns exemplos de vocabulários comuns são o *Dublin Core Metadata Initiative* (DCMI) [3] e o *E-Government Metadata Standard* (e-GMS) [12], ambos com semântica especializada para gerenciamento de conteúdo em aplicações de Governo Eletrônico.

4. *Avaliação do Processo* - Um dos desafios mais críticos da adoção de TIC está em medir o impacto de seu uso. Como avaliar a eficiência das ferramentas de TIC no processo político? Conforme o relatório da OECD [20], o investimento financeiro necessário para avaliar o uso de TIC é muito menor do que o investimento para o desenvolvimento de ferramentas. A maioria das pesquisas são realizadas com a intenção de se conhecer a satisfação do cidadão, enquanto que o impacto de suas contribuições no processo político não tem sido pesquisado e nem documentado. Esse desafio envolve não apenas questões tecnológicas, mas também uma mescla de aspectos técnicos, políticos e sociais.

5. *Garantia de Continuidade do Processo* - Para responder às expectativas dos cidadãos, é fundamental garantir a disseminação do processo político, demonstrá-lo na prática e validá-lo frequentemente, de modo que pequenas adaptações e evoluções sejam feitas.

Para ilustrar a complexidade desses problemas, tomemos o Brasil como exemplo, um país de dimensões continentais com 5560 municípios, segundo dados do IBGE de 2004/2005 [23]. Tais municípios estão distribuídos entre as regiões norte, com 449 municípios, nordeste com 1792, sudeste com 1668 (dos quais 645 apenas em São Paulo), região sul com 1188 e centro-oeste com 463 municípios. Esses dados evidenciam a necessidade de escalabilidade e coerência no acesso às informações publicadas, bem como a necessidade de integração em ambientes possivelmente heterogêneos e com total independência administrativa. Assim, não se espera que

uma única ferramenta seja capaz de prover o suporte adequado para esses diferentes tipos de relacionamentos. Pelo contrário, espera-se que um conjunto integrado de ferramentas e serviços seja utilizado.

3. GRADES COMPUTACIONAIS

Grades computacionais vêm surgindo como uma nova tecnologia para desenvolvimento de aplicações distribuídas, permitindo o compartilhamento e disponibilização de recursos sob demanda e em larga escala, através de um novo paradigma de computação baseado em serviços. Mais do que compartilhar recursos computacionais para aplicações paralelas que necessitam de alto poder de processamento, a adoção de arquiteturas para desenvolvimento de aplicações baseadas em serviços grid (*Grid Services*) permite a virtualização de recursos, encapsulando detalhes de implementação e criando um ambiente colaborativo e geograficamente distribuído [2, 8, 9].

As plataformas de grade permitem o compartilhamento de uma variedade de recursos, tais como servidores, estações de trabalho, computadores pessoais, dispositivos de armazenamento, ferramentas de software, etc., que podem estar em localizações geográficas dispersas e sob diferentes domínios administrativos.

Uma grade computacional possui características específicas, não presentes em outras plataformas distribuídas, dentre as quais destacam-se:

- *Escalabilidade e heterogeneidade.* Plataformas de grade podem crescer em número na casa de milhões de recursos, com heterogeneidade de hardware e software, diferentemente de outros ambientes computacionais distribuídos que agregam poucos recursos e tipicamente homogêneos, tais como clusters;

- *Dispersão geográfica.* Grades computacionais podem agregar recursos geograficamente distribuídos, atingindo uma escala global;

- *Múltiplos domínios administrativos.* Grades computacionais não possuem um administrador único. Cada instituição que oferece recursos para uma grade computacional tem total controle sobre sua utilização, definindo políticas de acesso, tanto a recursos quanto a serviços.

O novo paradigma de computação baseado em serviços aliado à tecnologia de grades computacionais, deu origem ao padrão *Open Grid Service Architecture* (OGSA) [10], especificado pelo *Global Grid Forum* (GGF). O padrão OGSA define mecanismos para implementação de serviços baseado em protocolos e padrões web, tais como *Simple Object Access Protocol* (SOAP) e *Web Service Description Language* (WSDL).

Atualmente, o Globus Toolkit 4 (GT4) [1] é a implementação de referência do padrão OGSA. O GT4 fornece um conjunto de bibliotecas e programas com objetivo de resolver problemas comuns na construção de sistemas distribuídos [6]. Alguns serviços são disponibilizados pelo GT4, tais como o gerenciamento e execução de tarefas, implementado pelo componente GRAM, os serviços de acesso a dados implementados pelos componentes GridFTP (transferência segura de dados) e RFT (gerenciamento confiável de múltiplas execuções do GridFTP), replicação de dados (RLS e DRS), e o acesso a bases de dados heterogêneas (OGSA-DAI). O GT4 oferece ainda um serviço de monitoramento e descoberta de recursos (WebMDS), e serviços de autenticação e controle de acesso a recursos e serviços na grade (implementados pelo GSI). Além dos serviços já implementados, o GT4 inclui suporte para desenvolvimento de novos serviços baseados no padrão *Web Services Resource Framework* (WSRF), [7].

Tanto em função dos desafios relacionados ao desenvolvimento de aplicações de democracia eletrônica, quanto da necessidade de estimular o engajamento e a participação ativa da sociedade no processo político, faz-se necessário a utilização de uma tecnologia que permita atender tais requisitos de forma trasparente, criando assim um ambiente integrado e colaborativo. Em [25], mostramos como uma plataforma de grade computacional pode ser utilizada para prover suporte para aplicações de democracia eletrônica.

4. DEMOCRACIA ELETRÔNICA EM GRADES COMPUTACIONAIS

4.1 Cenário

O conceito de democracia eletrônica é muito amplo, possibilitando o desenvolvimento de ferramentas para diversas etapas do ciclo político [15]. Nesse sentido, este trabalho propõe a criação de um conjunto de serviços capaz de dar suporte ao processo de elaboração de novas políticas públicas. Esses serviços são integrados e disponibilizados como uma ferramenta de fórum de discussões em um ambiente de grade. Na implementação destes serviços utilizamos o Globus Toolkit 4 (GT4) e seus componentes.

A Figura 1 ilustra o cenário de implementação da ferramenta de democracia eletrônica. De acordo com este cenário, diversos municípios de um país podem ser agrupados em grandes Áreas Metropolitanas (AM). Cada AM constitui uma Organização Virtual (OV) formada pela congregação de diversos municípios próximos que dispobibilizam recursos computacionais, e implementam e utilizam serviços de democracia eletrônica. Dessa forma, cada AM possuirá um ou mais servidores com o GT4 e outras ferramentas necessárias instaladas (servidor web, banco de dados, etc.). O acesso aos recursos será feito através de um portal Web.

Inicialmente, esta aplicação considera a participarão de três atores no ciclo político: *Mediador*, responsável pela administração de uma AM; *Parlamentar* (e.g., vereador, deputado,

[1] Globus Toolkit 4 website http://www.globus.org

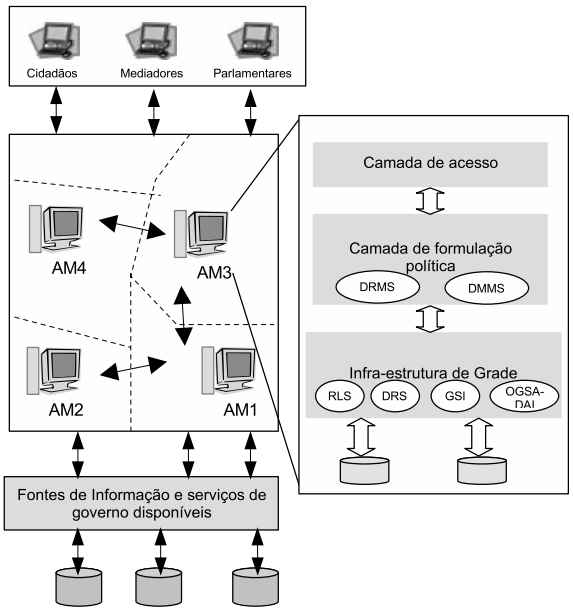


Figura 1: Cenário de implementação [25]

etc.), pode consultar opiniões, solicitar abertura de novos fóruns, publicar documentos, interagindo, assim, com a população no processo político; e *Cidadãos*, os quais podem ser cadastrados no sistema ou interagir como anônimos, sem necessidade de autenticação. Toda informação publicada será tratada como um recurso, podendo este ser uma mensagem, um tópico de discussão, um documento ou uma agregação de todos os outros recursos.

Nessa fase do projeto centramos esforços na definição e implementação de um conjunto de serviços que seja capaz de dar suporte a diferentes aplicações de democracia eletrônica, possibilitando sua utilização em larga escala. Os serviços implementados e apresentados neste artigo fazem parte da camada de formulação política. Nosso protótipo ainda disponibiliza um portal web com as funcionalidades básicas, tais como, publicação de mensagens e documentos, consultas e acompanhamentos de discussões, criação e encerramento de fóruns, etc.

A definição da arquitetura da ferramenta e seus aspectos de implementação serão detalhados a seguir. Para a implementação dos serviços foi necessário a definição de um conjunto de elementos de metadados, os quais foram detalhados em [25].

4.2 Colaboração em Larga Escala

Atualmente a Internet interconecta milhões de computadores no mundo inteiro. O seu crescimento em escala global possibilita o desenvolvimento de várias aplicações, dentre as quais estão os ambientes colaborativos. O grande volume de conteúdo publicado, bem como o potencial para integração de ferramentas de recuperação e troca de informações, são fundamentais para a criação de tais ambientes.

A democracia eletrônica apresenta novos desafios no que diz respeito à colaboração em larga escala na Internet. Espera-se que milhões de pessoas participem do processo político em países de dimensões continentais como o Brasil. Nesse sentido, ferramentas de TIC devem estar preparadas para dar suporte à colaboração, oferecendo mecanismos eficientes de descoberta e recuperação de informações em ambientes geograficamente distribuídos e, possivelmente, heterogêneos. Adicionalmente, tais ferramentas devem ser capazes de aproximar pesssoas através do compartilhamento de idéias e opiniões.

A ferramenta apresentada neste trabalho provê um fórum de discussão baseado no uso de plataformas de grade. As principais características da ferramenta e como ela atende aos requisitos de democracia eletrônica são apresentados a seguir.

Utilização em Larga Escala: Para explorar as características das plataformas de grade referente a escalabilidade, a ferramenta de democracia eletrônica deve ser executada no âmbito de cada Área Metropolitana (AM), a qual constitui uma Organização Virtual (OV) na grade computacional. Cada AM possui um servidor principal que implementa um serviço de informações, no qual são registrados todos os recursos e serviços disponíveis na mesma AM. Assim, usuários e serviços que executam na grade podem efetuar consultas a esse serviço para obter informações sobre a existência e o estado dos recursos de uma AM. No âmbito de cada AM, diversos repositórios são implementados para armazenar e disponibilizar recursos tais como, fóruns de discussão, mensagens e documentos publicados. Sempre que um novo recurso é criado, réplicas serão criadas e armazenadas em repositórios da mesma AM. O gerenciamento de réplicas é feito com o suporte do RLS, um componente do GT4 que implementa mecanismos para registro e localização de réplicas na grade computacional. A replicação, nesse caso, propicia maior disponibilidade dos recursos, permitindo que as aplicações se adaptem em caso de falhas.
Em algumas situações, aplicações precisam buscar e acessar recursos pertencentes a outras AMs. Para permitir buscas em AMs remotas, foi implementada uma estrutura hierárquica, na qual uma AM principal mantém registros sobre outras AMs associadas. Por exemplo, a AM principal pode ser mantida pelo Governo Estadual ou Federal, e manter registros sobre as demais AMs. Repositórios *cache* são implementados em cada AM, permitindo que os recursos remotos (i.e., obtidos de outras AMs) que são utilizados com mais frequência sejam obtidos com maior rapidez. O controle das réplicas de recursos remotos que estejam temporariamente armazenadas no *cache* local também são feitas pelo RLS. Um mecanismo de limpeza do cache local é disparado periodicamente, eliminando réplicas que não tenham sido utilizadas recentemente.

A Figura 2 demonstra o cenário de configuração do ambiente representando uma AM utilizada para testes. Nesta figura estão representados um servidor configurado para cache, dois servidores RLS para catálogo de réplica (LRC), um servidor RLS para localização de réplicas (RLI) e um servidor principal onde está localizado o *Portlet Container*, o serviço de informações (MDS) e a base de dados da aplicação.

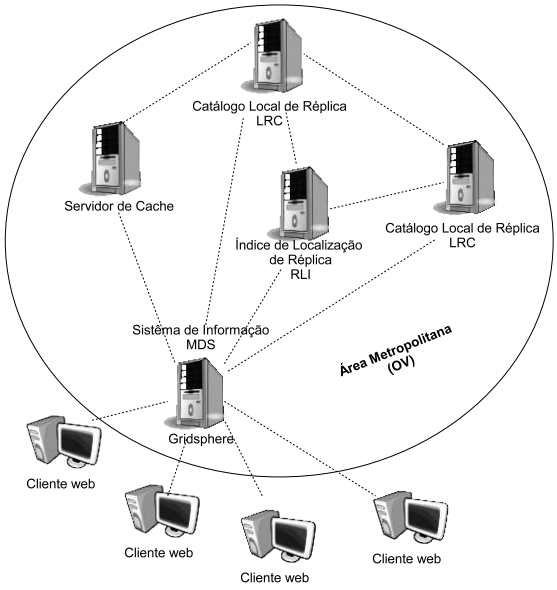


Figura 2: Cenário de configuração do ambiente

Orientação a Metadados: O protótipo descrito utiliza um metamodelo de metadados (inicialmente proposto em [25]) baseado nos padrões DCMI [3], e-GMS [12] e v-Card [27], sendo representados através de Resource Description Framework (RDF) [19]. Metadados permitem melhorar a acessibilidade e usabilidade de recursos, possibilitando maior agilidade e eficiência na descoberta de informações relevantes.

O metamodelo implementado permite que usuários possam adicionar novos elementos dinamicamente, conforme necessidade. Isto é feito criando-se novos pares de elementos chave/valor, onde a chave representa o nome do elemento acrescentado pelo usuário. Tal característica permite aos mediadores, por exemplo, agrupar informações (mensagens, documentos) e classificá-las com elementos específicos de metadados, tornando a busca pela informação mais eficiente.

Distribuição e Independência Administrativa: Grades computacionais diferenciam-se de outras plataformas distribuídas principalmente por permitirem independência adminstrativa e pertencerem a regiões geograficamente distantes. A ferramenta de democracia eletrônica explora essas características através da configuração de OV em cada Área Metropolitana. Cada OV terá instalado um portal web que permite ao mediador cadastrar usuários e grupos, atribuir permissões e gerenciar conteúdo publicado, agrupando e classificando informações. Informações localizadas em diferentes OVs poderão ser consultadas e recuperadas para a OV local através de implementação de serviços de replicação que fazem acesso aos componentes RLS e GridFTP disponibilizados pelo GT4.

Acessibilidade: O acesso às informações é feito através de um portal Web. O portal oferece mecanismos de fácil acesso e navegação. O portal utiliza *portlets* (veja seção 4.4), permitindo ao usuário personalizar sua área de trabalho através da seleção dos componentes que deseja acessar.

4.3 Arquitetura da Ferramenta

A arquitetura da ferramenta está baseada na utilização do Globus Toolkit 4 (GT4) e alguns de seus componentes, estendendo-os para implementação de detalhes específicos da aplicação em questão. A Figura 3 apresenta a arquitetura da ferramenta de democracia eletrônica.

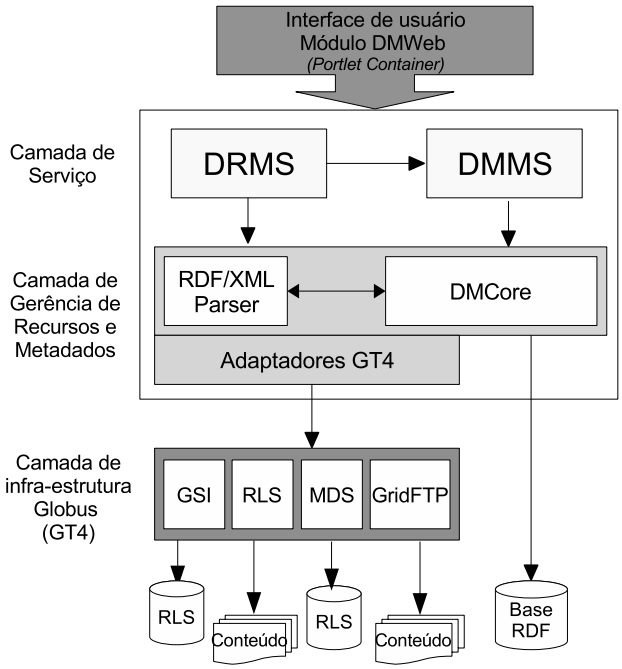


Figura 3: Arquitetura da ferramenta de democracia eletrônica

4.3.1 Infra-estrutura Globus

Na camada de mais baixo nível, denominada *camada de infra-estrutura do globus*, temos os componentes GT4 utilizados pela ferramenta de democracia eletrônica. Atualmente, os componentes utilizados são o GSI para autenticação, os componentes de acesso a dados RLS para replicação, GridFTP para recuperação de arquivos de conteúdo e o MDS para descoberta de serviços e recursos computacionais disponíveis.

A replicação tem por objetivo melhorar a disponibilidade da

aplicação e prover melhor qualidade na busca de documentos relevantes, disponibilizados em qualquer uma das OVs. O RLS é utilizado como mecanismo para registro e descoberta de réplica dos recursos de democracia eletrônica dentro da grade. A publicação de qualquer recurso envolve as etapas de: (i) *identificação*, atribuindo uma URI que o identificará unicamente independente de sua localização; (ii) *replicação*, onde é feita uma ou mais cópias do recurso; e (iii) *registro* dos mapeamentos das cópias em uma coleção de réplicas. As réplicas são feitas com o suporte do componentes GridFTP para transporte de arquivos de conteúdo, enquanto o RLS mantém registro dos recursos e de sua localização física.

4.3.2 Gerência de Recursos e Metadados

A camada intermediária, denominada *gerência de recursos e metadados (core)*, é responsável pela manipulação de metadados específicos da aplicação de democracia eletrônica, pelas estratégias de replicação e *cache* implementados e pelo acesso aos serviços de infra-estrutura do GT4. Essa camada implementa o modelo de metadados apresentado em [25].

O componente *DMCore* encapsula a manipulação e persistência de documentos RDF/XML, ocultando da camada de serviços os detalhes de implementação. Nessa mesma camada tem-se o componente *XML/RDF Parser*, que é responsável pela transformação de recursos e metadados em documentos XML e por gerar o grafo de persitência RDF para banco de dados relacional. Atualmente utilizamos o *framework* Jena 2.3[2] e o banco de dados Postgres para persistência de conteúdo RDF.

O componente *Adaptador GT4* é responsável por prover acesso aos componentes do Globus Toolkit 4. Esse componente permite encapsular o acesso à infra-estrutura da grade.

O componente DMCore permite implementar diferentes estratégias de replicação, que podem ser configuradas durante a implantação da aplicação, bastando apenas que uma nova estratégia implemente a interface `ReplicationStrategy`. A estratégia padrão para replicação seleciona dois LRCs aleatoriamente para criar as cópias. Esse componente é baseado nos padrões *Facade*, *AbstractFactory* e *Strategy* [11].

4.3.3 Serviços

A camada de mais alto nível, denominada *camada de serviços*, provê gerenciamento de sessão e interação com diferentes tipos de usuários. Nessa camada estão implementados os serviços necessários para dar suporte à execução da aplicação de democracia eletrônica em ambiente de grade. Os seguintes serviços foram implementados como *Grid Services* utilizando o padrão WSRF:

- *e-Democracy Resource Management Service* (DRMS) - serviço de acesso para manipulação de recursos na aplicação de democracia eletrônica.
- *e-Democracy Metadata Management Service* (DMMS) - serviço responsável pela manipulação de metadados específicos da aplicação e pela atribuição aos recursos.

[2]Semantic Web Framework, http://jena.sourceforge.net/

A. DRMS

O DRMS provê suporte à manipulação dos recursos de democracia eletrônica, como fóruns de discussões, mensagens, documentos e opiniões publicadas pelos atores envolvidos. Com o suporte do DRMS, mediadores podem criar e encerrar tópicos de discussão, publicar documentos relevantes para acesso pelos demais atores, aprovar ou rejeitar mensagens publicadas e agrupar e direcionar as mensagens a pessoas apropriadas da administração pública. Através do DRMS, cidadãos podem consultar informações e documentos previamente publicados e enviar suas próprias opiniões sobre os temas em debate, identificando-se ou não; o DRMS provê mecanismo para cadastramento e identificação de usuários, apenas quando esses queiram se identificar.
As funcionalidades suportadas pelo DRMS incluem:

- *Publicação de recursos.* O DRMS fornece funcionalidades específicas para publicação de recursos, tais como mensagens e documentos, bem como criação de tópicos de discussão. Parlamentares podem solicitar aos mediadores a abertura de novos fóruns e, cidadãos cadastrados na aplicação podem sugerir outros temas para discussão, ambos acessados através do serviço DRMS.
- *Recuperação de informações.* O DRMS, através da interação com o serviço DMMS, é acessado pelos atores para obtenção de informações específicas armazenadas em bases de dados replicadas.

B. DMMS

Todo recurso publicado na aplicação de democracia eletrônica será associado a um conjunto de informações de metadados. O DMMS provê suporte para o armazenamento e a consulta de metadados associados aos recursos publicados na aplicação. Semelhante a outros serviços de metadados [4], o DMMS provê um mecanismo para gerenciamento e processamento de metadados especializado para o contexto de democracia eletrônica. Nesse sentido, o DMMS implementa um serviço de catálogo de metadados que suporta armazenamento, consulta, classificação e agregação de recursos em larga escala. Com o suporte do DMMS, mediadores podem contribuir para melhorar a qualidade da informação disponível on-line, em termos de acessibilidade, relevância e utilidade para os cidadãos e parlamentares, promovendo uma melhor comunicação entre cidadãos, parlamentares e órgãos da administração pública.

O DMMS pode ser acessado pelo DRMS no momento da publicação de recursos de modo a atribuir os elementos mínimos de metadados informado pelos usuários. Outra importante característica do DMMS é a possibilidade dos usuários criarem seus próprios elementos de metadados. Através do DMMS, mediadores podem criar agrupamento de informações e atribuir novos elementos de metadados para

permitir uma melhor qualificação dos dados. Por exemplo, um mediador pode criar um agrupamento de recursos do tipo mensagem e atribuir um elemento de metadados chamado `<escolaridade>`, cujo valor do elemento seja `graduado`. Dessa forma, uma consulta poderia ser feita na base de dados RDF do tipo `selecione as mensagens publicadas pelos cidadãos graduados`. Essa característica acrescenta um valor significativo ao conteúdo publicado, sendo o DMMS flexível o suficente para permitir evoluções no mecanismo de armazenamento e recuperação de informações armazenadas em RDF/XML.

4.4 Protótipo da Ferramenta

A ferramenta de democracia eletrônica é disponibilizada através de um portal Web, cuja implementação está baseada em *portlets*. *Portlets* são componentes Web com um conjunto bem definido de operações, os quais fornecem interfaces visuais padronizadas e reutilizáveis [13]. Da mesma forma que um *Web Container* disponibiliza um ambiente para execução de aplicações Web, um *Portlet Container* gerencia a execução e o ciclo de vida de um *Portlet*. Nosso protótipo utiliza o *Portlet Container GridSphere*[3]. O *GridSphere* fornece um conjunto básico de componentes necessários para o gerenciamento do portal, como Login/Logout, gerenciamento de contas e gerenciamento de usuários. O *Portlet Subscription* permite que o próprio usuário selecione os componentes aos quais deseja utilizar, conforme regras de acesso definidas pelo administrador. A Figura 4 apresenta a tela de gerenciamento de permissões aplicado a cada *portlet*. Existem dois grupos de *portlets*, o primeiro representa o grupo dos componentes para aplicações de grade em geral (*Grid Portlets*), e o segundo, os componentes específicos da aplicação de democracia eletrônica, cada um com seu nível de permissão; nesse caso, usuários podem acessar os componentes para acompanhamento de discussões, publicação de mensagens e publicação de documentos. Outros agrupamentos podem ser criados, inclusive com um *portlet* participando de mais de um grupo.

Subscribe	gridportlets	Portlet description	Required role
☐	Job Submission Portlet	Job Submission Portlet	User
☐	File Activity Portlet	File Activity Portlet	User
☐	Resource Browser Portlet	Resource Browser Portlet	User
☐	File Browser Portlet	File Browser Portlet	User
☐	Credential Manager Portlet	Credential Manager Portlet	User
☐	Resource Registry Portlet	Resource Registry Portlet	User
Subscribe	**Democracia Eletronica**	**Portlet description**	**Required role**
☑	Acompanhar discussões	Acompanhar discussões	User
☑	Gerenciamento de Metadados	Gerenciamento de Metadados	Super
☑	Publicar Mensagem	Publicar Mensagens	User
☑	Gerenciar Topicos de Discussao	Gerenciar Topicos de Discussao	Super
☑	Publicar Documentos	Publicar Documentos	User

Save | Cancel

Figura 4: Gerenciamento de permissões do *Gridsphere*

Adicionalmente, o GridSphere fornece um conjunto de componentes denominados *Grid Portlets* [24] (veja o primeiro grupo de componentes da Figura 4). Tais componentes suportam acesso ao Globus Toolkit 2 (GT2), GT3 e GT4, oferecendo interface para acesso aos serviços de localização, submissão de tarefas, FTP, gerenciamento de credencias, autenticação, entre outros.

A camada de acesso aos serviços da ferramenta estão encapsulados em um módulo Web denominado DMWeb. O DMWeb implementa componentes *portlets* atômicos, ou seja, cada *portlet* disponibiliza uma funcionalidade específica, permitindo uma granularidade mais fina no gerenciamento de acesso aos serviços. A Figura 5 apresenta a interface da ferramenta de democracia eletrônica. Nesta interface o usuário pode acompanhar as discussões, consultando as informações publicadas apartir de palavra-chave, tópicos específicos ou elementos de metadados.

5. TRABALHOS RELACIONADOS

Atualmente, existem várias formas de interação entre grupos e espaços para discussões de assuntos de interesse geral. Um dos meios mais utilizados, e que vem crescendo gradativamente, é a disponibilização de *Weblogs* [16]. Não somente o usuário "comum" da internet pode disponibilizar seu *Weblog* para expor suas opiniões, como muitos veículos de comunicação, ONGs, jornalistas, parlamentares, etc. publicam suas idéias e opiniões em tais ambientes, permitindo que qualquer pessoa interaja e discuta suas opiniões. Além dos *Weblogs* e ferramentas de fórum de discussões de interesse geral, algumas iniciativas veêm sendo tomadas no sentido de disponibilizar ferramentas específicas para aplicações de Democracia Eletrônica e criação de ambientes colaborativos através da internet.

O projeto *Webocracy* [17] criou a ferramenta denominada *Webocrat* para assegurar suporte eficiente à troca de informações entre cidadãos e governos. Além de oferecer módulos como gerenciador de fóruns de discussão, gerenciamento de conteúdo web, consulta e obtenção de documentos e geração de relatórios, todos acessados através da internet, o *Webocrat* está centrado na implementação do *Knowledge Model* (KM), responsável por gerenciar ontologias e dar suporte a todos os outros módulos da aplicação. O módulo KM implementa uma tecnologia de modelagem de conhecimento baseado na organização de informações utilizando um modelo específico (*domain specific knowledge model*), o qual resulta em uma eficiente capacidade de recuperação de informações.

O projeto DEMOS [29] é um sistema disponibilizado através da plataforma *Wornex's WorldDirector* que provê ambiente para gerenciamento de discussões e tomadas de decisão através da Internet, suportado por um conjunto de ferramentas integradas e adaptadas para múltiplas linguagens. O projeto foi validado a partir da utilização em Hamburgo (Alemanha) e Bologna (Itália). O sistema é disponibilizado através do modelo *Application Service Providers (ASP)*, o qual possibilita a locação da aplicação, ao invés da aquisição.

[3]Gridsphere website http://www.gridsphere.org

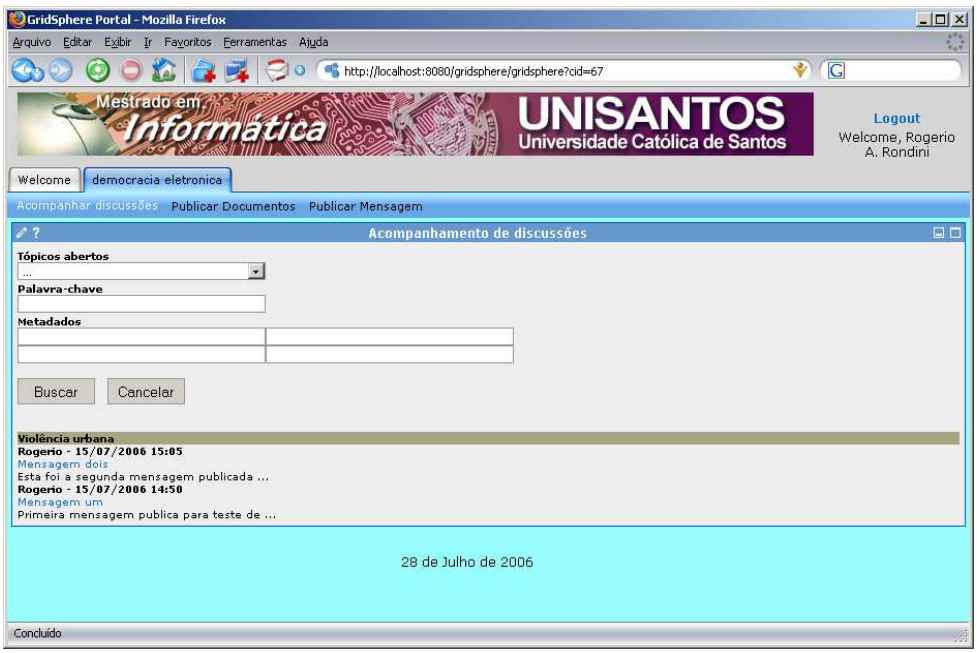


Figura 5: Interface de acompanhamento de discussões

O projeto EURO-CITI [5] explora o potencial da democracia eletrônica através da utilização de serviços como voto eletrônico, submissão de formulários e documentos e com consultas aos cidadãos (*e-consulting*). O sistema oferece acesso através da Internet ou quiosques localizados em lugares públicos.

O projeto CoFrame [14] implementa um *framework* para desenvolvimento de aplicações colaborativas, baseado em *Grid Services* e no Globus Toolkit 3 (GT3). O CoFrame foi utilizado para implementar, como estudo de caso, um sistema de educação a distância na China, chamado *China Central Radio e TV University* (CCRTVU). Adicionalmente, o CoFrame implementa mecanismos de replicação de informações baseados em três diferentes estratégias: (i) volume de tráfego na rede, (ii) carga dos servidores e (iii) popularidade dos recursos acessados.

Nosso trabalho assemelha-se ao Webocracy quanto à concepção da ferramenta e entendimento dos requisitos de democracia eletrônica. Ambos procuram implementar mecanismos de busca e classificação de informações eficientes. O trabalho também possui objetivos similares ao projeto CoFrame, dado que ambos procuram prover ferramentas que atendam aos requisitos de escala, heterogeneidade, distribuição geográfica e compartilhamento de informações para construção de ambientes colaborativos de larga escala.

6. AVALIAÇÃO DE RESULTADOS

De modo geral, os potenciais benefícios advindos do uso de tecnologias colaborativas incluem a melhoria da qualidade nas deliberações, maior transparência no processo político, e uma participação mais ampla dos cidadãos [20, 21, 22]. Entretanto, pouco se sabe sobre o impacto efetivo do uso de tais ferramentas colaborativas no ciclo político.

Avaliar os benefícios da democracia eletrônica pode ser bem mais complexo do que avaliar a acessibilidade ou aceitação de aplicações que implementam fóruns de discussão na Web. Por exemplo, tais ferramentas podem envolver diversos canais de interação *online* e *offline* entre cidadãos, seus representantes eleitos, e entidades governamentais ou não-governamentais. A adequação das ferramentas à forma como os processos políticos ocorrem na prática também constitui um fator importante a ser observado. A mediação da interação entre os atores envolvidos no ciclo político também é de fundamental importância para que os benefícios possam ser atingidos, e a questão envolve um estudo aprofundado sobre a forma como as pessoas se comportam no processo político (por exemplo, formando parcerias, oposições, *lobbies*, etc). Esses e outros aspectos técnicos e sociais têm sido amplamente discutidos na literatura e fóruns especializados em Sistemas Colaborativos. Sistemas de software cujo objetivo envolve o estabelecimento ou facilitação da conversação, a formalização dos papéis dos atores envolvidos em uma interação, o processamento das suas interações, ou ainda, a provisão de outras formas de suporte têm sido objeto de estudo há mais de uma década.

Apesar desses aspectos serem amplamente estudados há vários anos, novas oportunidades de pesquisa surgem quando tais aspectos são trazidos para o contexto da democracia

eletrônica. Nesse sentido, uma das principais dificuldades atuais é a carência de experimentos e relatos mais completos e aprofundados, efetivamente ligados a processos políticos existentes, principalmente envolvendo um número expressivo de atores [1]. Por esses motivos, mais recentemente a democracia eletrônica tem ganhado importância crescente nos fóruns especializados em sistemas colaborativos [28].

7. CONCLUSÃO E TRABALHOS FUTUROS

O sucesso do uso de TIC em democracia eletrônica depende da criação de espaços públicos para discussão, comunicação interativa entre os atores envolvidos, garantia da qualidade da informação, permitindo busca, classificação e agregação de forma eficiente e, garantia de que realidades específicas de cada região geográfica sejam refletidas no espaço público bem como a interação entre cidadão de regiões geograficamente distribuídas.

Uma das principais limitações das ferramentas colaborativas, tais como *weblogs* e fóruns de discussão, está no tratamento da grande quantidade de informação publicada, que se perde com o tempo. Para que exista cooperação entre pessoas e compartilhamento de informações, principalmente no que se refere à democracia eletrônica, é fundamental que existam mecanismos para localização e classificação eficientes, de forma que as opiniões publicadas sejam ouvidas durante o processo de elaboração de políticas públicas.

Nesse sentido, este trabalho procura ocupar a lacuna deixada pelos atuais ambientes públicos de colaboração. Plataformas de grade oferecem capacidade de gerenciamente de recursos e informações em ambientes geograficamente distribuídos e heterogêneos, características fundamentais para cooperação em larga escala. Através dos recursos oferecidos pelas plataformas de grade, especificamente os ambientes baseados em serviços (*Grid Services*), torna-se possível o desenvolvimento de aplicações colaborativas, possibilitando maior compartilhamento e disseminacão de informações.
Com o protótipo atual da ferramenta, podemos assegurar uma maior integração dos recursos existentes nas diversas AMs. Se cada AM disponibilizasse apenas um *Web Service* comum, teríamos apenas "ilhas de recursos", isoladas umas das outras. Isso certamente limitaria a escala de colaboração. Com a utilização de Grades computacionais, é possível integrar recursos disponíveis em diversas AMs.

Os próximos passos deste projeto envolvem a realização de experimentos com a implementação atual dos serviços de gerência de recursos. O objetivo principal de tais experimentos será avaliar a eficiência na busca de informações com a utilização de semântica RDF como técnica para melhorar a acessibilidade do cidadão aos recursos publicados. Pretende-se ainda, implementar e avaliar um serviço de gerência de identidade e controle de acesso a recursos, aplicados ao contexto de democracia eletrônica, de modo a ampliar as características já disponibilizadas pelos *Portlets Containers*. Adicionalmente, serão avaliados aspectos de acessibilidade das informações e usabilidade da ferramenta. Por fim, pretende-se disponibilizar a ferramenta para entidades interessadas na sua utilização em situações reais, de forma que os aspectos sociais e políticos possam ser avaliados.

8. REFERÊNCIAS

[1] S. Coleman and J. Gøtze. Bowling together: Online public engagement in policy deliberation. Home Page em http://bowlingtogether.net/bowlingtogether.pdf, 2001. (acessado em setembro de 2006).

[2] K. Czajkowski, C. Kesselman, S. Fitzgerald, and I. Foster. Grid information services for distributed resource sharing. In *10th IEEE International Symposium on High Performance Distributed Computing*, Los Alamitos, CA, USA, 2001. IEEE Computer Society Press.

[3] DCMI. Dublin core metadata initiative dcmi. Home page http://dublincore.org/, 2005. (acessado em setembro de 2006).

[4] E. Deelman, G. Singh, M. P. Atkinson, A. Chervenak, N. P. C. Hong, C. Kesselman, S. Patil, L. Pearlman, and M.-H. Su. Grid-based metadata service. In *16th Intl. Conf. on Scientific and Statistical Database Management*, page 393, Santorini, Island, Greece, 2004. IEEE Computer Society Press.

[5] EURO-CITI. European cities platform for on-line transaction services. Home page http://www.euro-citi.org/, 1999. (acessado em setembro de 2006).

[6] I. Foster. Globus toolkit version 4: Software for service-oriented systems. In *IFIP International Conference on Network and Parallel Computing*, pages 2–13. Springer-Verlag LNCS, 2005.

[7] I. Foster, J. Frey, S. Graham, S. Tuecke, K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, I. Sedukhin, D. Snelling, T. Storey, W. Vambenepe, and S. Weerawarana. Modeling stateful resources with web services v. 1.1., março 2004.

[8] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. In *Open Grid Service Infrastructure WG*. Global Grid Forum, junho 2002. (extended version of Grid Services for Distributed System Integration).

[9] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Super-Computer Applications*, 15(3), 2001.

[10] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. V. Reich. The open grid services architecture, version 1.0. Global Grid Forum (GGF), janeiro 2005. Informational Document.

[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995.

[12] GovTalk. e-government metadata standard (e-gms) v. 3.0. Home page http://www.govtalk.gov.uk/schemasstandards/metadata_document.asp?docnum=872, abril 2004. (acessado em setembro de 2006).

[13] Java Community Process. Jsr 168: Portlet specification. Home page http://www.jcp.org/jsr/detail/168.jsp, 2003. (acessado em setembro de 2006).

[14] J. Jiang, S. Zhang, Y. Li, and M. shi. Coframe: A framework for CSCW applications based on grid and web services. In *IEEE Intl. Conf. on Web Services*, pages 570–577. IEEE Computer Society Press, 2005.

[15] H. Kubicek, H. Westholm, and R. Winkler. Prisma strategic guideline 9. Home page http://www.prisma-eu.net/deliverables/sg9democracy.pdf, abril 2003. (acessado em setembro de 2006).

[16] C. Lindahl and E. Blount. Weblogs: Simplifying web publishing. *Computer*, 36(11):114–116, Novembro 2003.

[17] M. Mach, T. Cizmarik, F. Dridi, K. Furdik, J. Gilberg, J. Hreno, S. Hudak, R. Kende, M. Lumtzer, P. Macej, E. Novacek, I. Palola, T. Sabol, N. Sansom, and P. Thomson. Webocracy - edited annual report for publication. Technical report, Information Society Technologies, dezembro 2003.

[18] A. Macintosh. Characterizing e-participation in policy-making. In *37th Hawaii Intl. Conf. on System Sciences*, page 50117a, Los Alamitos, CA, USA, 2004. IEEE Computer Society Press.

[19] B. McBride. Rdf primer - w3c recommendation. Home page http://www.w3.org/TR/rdf-primer/, fevereiro 2004. (acessado em setembro de 2006).

[20] OECD. Citizens as partner information, consultation and public participation in policy-making. Home page http://www.oecd.org/dataoecd/53/57/2537528.pdf, 2001. (acessado em setembro de 2006).

[21] OECD. Engaging citizens online for better policy-making. policy brief. Home page http://www.oecd.org/dataoecd/62/23/2501856.pdf, 2003. (acessado em setembro de 2006).

[22] OECD. *Promise and Problems of e-Democracy. Challenges of online Citizen Engagement*. OECD Publications Service, Paris - France, 2003. Home page http://www.oecd.org/publications/e-book/4204011E.pdf (acessado em setembro de 2006).

[23] L. A. P. Oliveira. Perfil dos municípios brasileiros: Gestão pública 2004/IBGE. Pesquisa de Informações Básicas Municipais. ISBN 85-240-3837-3, IBGE - Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro, 2005. Coordenação de População e Indicadores Sociais.

[24] M. Russell, J. Novotny, and O. Wehrens. The grid portlets web application: A grid portal framework. GridSphere Project Website - http://www.gridsphere.org/gridsphere/gridsphere?cid=publications, 2005. (acessado em setembro de 2006).

[25] H. Senger, F. Silva, M. Mendes, R. Rondini, and C. Farias. Grid platforms for e-democracy applications. In *11th IEEE Symposium in Computer and Comunications*, pages 334 – 339, Pulla, Cagliari, Junho 2006. IEEE Computer Society Press.

[26] F. Silva and H. Senger. The grid: an enabling infrastructure for future e-commerce, e-business and e-government applications. In *Digital Communities in a Networked Society: eCommerce, eGovernment and eBusiness*, pages 253 – 265, Dordrecht - Holanda, 2004. Kluwer Academic Publishers.

[27] Versit. vcard: The electronic business card v. 2.1 - a versit consortium specification. Home page http://www.imc.org/pdi/vcardwhite.html, setembro 1996. (acessado em setembro de 2006).

[28] A. Whyte. Conversations with non-human actors in e-democracy evaluation. In *ECSCW e-Democracy Workshop*, Helsinki, Finland, 2003. ECSCW e-Democracy Workshop.

[29] Wornex. Delphi online mediation system. Home page http://www.wornex.com/content/view/16/83/, 2002. (acessado em setembro de 2006).

Simbiose Digital e Colaboração em Ambientes 3D

Glauco Todesco
Universidade de Sorocaba
Rod. Raposo Tavares, KM 92.5
Sorocaba – São Paulo – Brasil
Telefone: (+55 15) 2101-7000
glauco.todesco@uniso.br

Marcelo K. Zuffo
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158. Trav.3.
São Paulo - SP – Brasil
Telefone (+55 11) 3091-5589
mkzuffo@lsi.usp.br

RESUMO
Esse artigo discute sobre as atuais formas de colaboração em ambientes tridimensionais e propõe um novo paradigma denominado de Simbiose Digital com suporte a colaboração baseada na experimentação sensorial coletiva.

ABSTRACT
This paper discusses about the actual forms of collaboration in the three-dimensional environments and proposes a new paradigm called Digital Symbiosis that supports collaboration based on collective sensorial experience.

Palavras Chaves
Simbiose Digital, Trabalho Cooperativo, Realidade Virtual e Realidade Aumentada.

KeyWords
Digital Symbiosis, Cooperative Work, Virtual Reality and Argumented Reality.

1. INTRODUÇÃO
Desde o surgimento dos meios de comunicação e dos computadores, as possibilidades e formas de colaboração aumentaram exponencialmente. Entre essas formas de colaboração, esse trabalho discute sobre colaboração em ambientes tridimensionais. Todas as formas de colaboração baseadas em ambientes tridimensionais geralmente partem do principio de manter a individualidade de cada um, principalmente nas aplicações baseadas em ambientes virtuais colaborativos, sistemas de tele-presença e ambientes de realidade aumentada colaborativos. Manter essa individualidade, onde cada um decide o que vê, com que se relacionada ou que faz, é um reflexo do que acontece no mundo real e, muitas vezes, manter esse tipo de relacionamento com os outros participantes e com o ambiente é interessante. Porém uma área ainda não explorada em ambientes tridimensionais com suporte a colaboração são aquelas onde a experimentação sensorial é coletiva, ou seja, a experiência sensorial de um participante é a mesma para todos os participantes, sendo essa a principal característica de uma aplicação de Simbiose Digital.

SBSC Simpósio Brasileiro de Sistemas Colaborativos
20 a 22 de Novembro de 2006
Natal, RN, Brasil

© Sociedade Brasileira de Computação
Comissão Especial de Sistemas Colaborativos

Anais do III Simpósio Brasileiro de Sistemas Colaborativos, pp. 161-171.

Esse artigo discute inicialmente as formas tradicionais de colaboração baseadas em ambientes tridimensionais. Após essa discussão é apresentada a definição de Simbiose Digital, bem como as suas principais características. Um exemplo de aplicação de Simbiose Digital é apresentado seguido das conclusões.

2. COLABORAÇÃO EM AMBIENTES 3D
Segundo Benford et al [1], as aplicações de CSCW podem ser classificadas espacialmente em cinco categorias: Espaços de Mídia, Vídeo Conferência Espacial, Ambiente Virtual Colaborativo, Tele-Presença e Ambiente de Realidade Aumentada Colaborativo. A dimensão espacial está relacionada ao nível de suporte das principais propriedades físicas espaciais como topologia, distância, orientação e movimento. Os espaços de mídias envolvem a melhoria dos ambientes de trabalho com a integração da comunicação baseada no uso de áudio e vídeo. As principais características desse tipo de aplicação estão baseadas em tentar reunir pessoas em regiões geográficas distintas por um longo período de tempo, focando em aspectos sociais de relacionamento, sensação de presença e consciência periférica. A vídeo conferência tradicional envolve o uso de vídeo e áudio para suportar encontros, geralmente de curta duração e com propósitos pré-definidos. As aplicações de Vídeo Conferência Espacial são uma variação das aplicações tradicionais de vídeo conferência, porém elas permitem que os usuários tenham a referência espacial do ambiente compartilhado, como permitir saber quem está olhando para quem (*gaze direction*). Os Ambientes Virtuais Colaborativos são aplicações baseadas no uso de um sistema de comunicação para o compartilhamento de ambientes sintéticos 3D. Os usuários são representados nesses ambientes virtuais através da figura de um avatar, podendo interagir com o ambiente e com os demais avatares de forma independente. A tele-presença permite que os participantes da aplicação possam remotamente estar presentes, através do uso de computadores e tecnologias de comunicação, permitindo que os usuários naveguem pelo ambiente ou ainda manipulem objetos reais remotamente. As aplicações de tele-presença são geralmente realizadas através do uso de um robô que realiza a captura de imagens que podem ser visualizadas através de um monitor simples, ou ainda através de dispositivos que permitem a imersão no ambiente remoto. Geralmente o uso da tele-presença se aplica em regiões onde existam riscos de vida ou ainda em casos especiais como as viagens espaciais. Os ambientes colaborativos de realidade aumentada envolvem sobrepor objetos gráficos dentro do mundo real, como por exemplo, em ambientes cirúrgicos onde as imagens gráficas indicam o local de incisão do bisturi no paciente. Os ambientes colaborativos de realidade aumentada focam o compartilhamento de um mesmo ambiente físico, permitindo que vários usuários possam colaborar na execução de uma tarefa. Cada tipo de aplicação varia entre o nível de artificialidade, que

pode ser desde o ambiente físico até o sintético; e o nível de transporte que está relacionado na localização remota ou local de cada usuário (Figura 1).

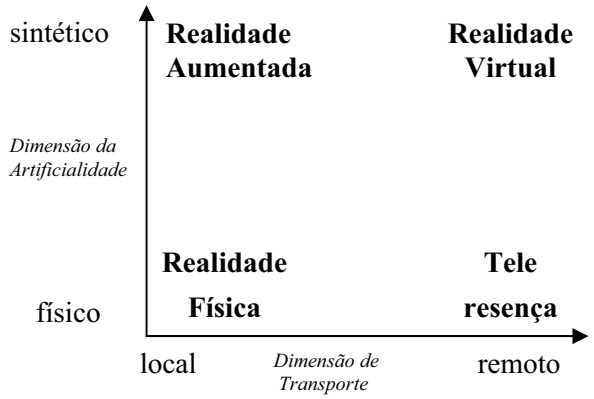


Figura 1 - Classificação Espacial - Benford et al (1998)

2.1 Realidade Mista

A Realidade Mista - RM (*Mixed Reality*) é definida por Milgram et al [21] como uma fusão dos mundos reais e virtuais em uma mesma aplicação. Milgram et al (1994) definiram uma taxonomia para RM, classificando diferentes tecnologias dentro de um espectro de acordo com o conceito de realidade-virtualidade continuada. A Realidade Mista refere-se à classe das aplicações que cobrem todos os dispositivos de visualização, combinando elementos do mundo real e virtual necessariamente. A RA está relacionada ao lado esquerdo do espectro, onde o contexto da aplicação está associado principalmente ao mundo real, sendo que esse mundo é "aumentado" com o uso de elementos gráficos gerados em tempo real. O lado direito do espectro está relacionado com o uso de elementos gráficos, onde o mundo virtual é "aumentado" através da adição de elementos ou sensações do mundo real. Em algumas aplicações, essa separação é bastante clara, porém em alguns casos, essa classificação não é trivial, principalmente em ambientes aonde o usuário pode se locomover livremente, o que termina por levar o usuário aos extremos do espectro durante a navegação, tornando a classificação da aplicação como Realidade Mista mais apropriada.

Benford et al [1] definiram Realidade Mista baseada no conceito do suporte à consciência e a comunicação entre os habitantes de diferentes espaços, definindo o conceito de fronteiras transparentes entre o mundo real e virtual. Podem existir várias fronteiras em uma mesma aplicação, onde o suporte à consciência e à comunicação entre os participantes deve ser o mais natural possível. Aplicações colaborativas em ambientes de realidade mista, como o proposto por Billinghurst e Kato [2], são também encontradas na literatura. Nesse tipo de aplicação dois ou mais usuários compartilham o mesmo ambiente físico e colaboram para a execução de alguma tarefa, através do uso dos conceitos de realidade aumentada.

2.2 Tele Ator

Goldberg et al [7] propuseram uma interface de votação dinâmica voltada para ambientes com vários operadores e um único robô. Esses ambientes podem ser classificados como tele-presença coletiva. A aplicação coleta os votos dos participantes que elegem a próxima ação do robô ou ser humano (Tele-Ator), que possui uma câmera que permite compartilhar a visão com os operadores remotos. Na arquitetura proposta, os usuários compartilham o controle do ator remoto através de um navegador WEB. Toda comunicação é concentrada em um servidor e existe a figura de um diretor, responsável em intermediar os usuários com o tele ator. O diretor se comunica com o tele-ator através de uma interface sem fio (voz). As imagens capturadas pelo ator remoto são transmitidas aos usuários e o diretor define o que será votado. Através da interface da aplicação, os usuários possuem um tempo determinado para votar, através de um clique em uma pessoa, objeto ou opções do tipo "sim" ou "não". Assim que o tempo de votação é esgotado, o resultado é conhecido pelo diretor e esse transmite ao tele ator e aos participantes.

2.3 Conclusões

Todas as formas de colaboração discutidas nesse tópico, exceto o Tele Ator, partem geralmente do principio em manter a individualidade de cada participante, ou seja, a experiência sensorial que cada participante é submetido durante a aplicação geralmente é individual. Manter essa individualidade, onde cada um decide o que vê, com que se relacionada ou que faz, é um reflexo do que acontece no mundo real e, muitas vezes, manter esse tipo de relacionamento com os outros participantes e com o ambiente é interessante e fundamental. Porém uma outra forma de colaboração em ambientes tridimensionais não discutida na literatura são as que estão baseadas na experimentação sensorial coletiva. Essa mudança em como ocorre a experimentação sensorial muda completamente o paradigma de colaboração entre os participantes de uma aplicação. No próximo tópico é apresentado e definido o termo Simbiose Digital, que possui como principal característica a experimentação sensorial coletiva.

3. SIMBIOSE DIGITAL

Segundo o Happer [10], o termo simbiose advém do grego e é formado pela junção dos radicais **sin** (junto) e **bios** (vida), e significa "Vida em Conjunto". Em 1877 esse termo foi utilizado pela biologia para definir a associação estreita entre dois ou mais seres vivos de espécies distintas, que estabelecem essa relação com benefícios mútuos (mutualismo) [10]. Dentro da psicologia, o termo simbiose é designado para determinar o relacionamento entre uma mãe e o seu bebê[1] e ainda possui o sentido figurado de associação íntima entre duas pessoas [11]. Duas ou mais pessoas podem estabelecer uma relação de simbiose para desenvolver alguma atividade ou ter algum benefício. Geralmente essa relação está baseada em pessoas que possuam afinidades ou um alto grau de entrosamento. Essa afinidade ou nível de entrosamento é, na maior parte das vezes, resultado de um grande tempo de convivência entre essas pessoas. O termo simbiose foi inicialmente aplicado na biologia, mas pode ser aplicado em outros contextos. O termo simbiose foi citado pela primeira vez dentro do contexto computacional em 1960 por Licklider [18]. Ele descreveu um relacionamento entre o operador (homem) e um computador, que são dois sistemas distintos, porém interdependentes. Homem e máquina cooperam entre si com o objetivo de resolver alguma tarefa, onde cada componente possui as suas próprias habilidades. O componente humano é mais indicado para tarefas que requerem criatividade como a elaboração de questões importantes e a tomada de decisões

[1] http://www.mercksource.com

críticas. O computador por sua vez é mais apropriado para a execução de tarefas que manipulam uma grande quantidade de dados (armazenamento e recuperação) e que necessitem de respostas rápidas e precisas. Os operadores humanos e os sistemas de computadores podem estabelecer um relacionamento de simbiose que permite um aumento na capacidade em realizar tarefas complexas.

3.1 Definição de Simbiose Digital

Nesse trabalho Simbiose Digital [28] é definida como uma forma estreita de associação estabelecida entre os seres humanos e os computadores, através dos meios eletrônicos interativos, com o objetivo de compartilhar as faculdades humanas e computacionais. Entendemos por meios eletrônicos interativos como o acervo tecnológico orientado ao relacionamento sensitivo multimodal entre o usuário e uma infraestrutura computacional [32]. O termo faculdade possui vários significados, entre eles, segundo o dicionário Houaiss [11], o termo faculdade é definido como a possibilidade, natural ou adquirida, de fazer qualquer coisa, tendo como sinônimos as palavras capacidade ou poder. Nesse artigo, o termo faculdade é usado como a capacidade de fazer qualquer coisa, e tanto o homem como o computador, possuem faculdades ou capacidades que podem ser compartilhadas. Na biologia, uma relação de simbiose é uma forma estreita de associação entre os simbiontes, por existir um relacionamento baseado no contato físico entre eles. Nesse artigo, entendemos por forma estreita de associação os relacionamentos baseados no contato físico e na comunicação entre os participantes a qualquer momento. Esse estreitamento também é caracterizado pela característica encontrada em uma aplicação de Simbiose Digital que é a experimentação sensorial coletiva discutido a seguir.

3.2 Faculdades

As faculdades envolvidas em uma aplicação de Simbiose Digital podem ser classificadas em dois grupos: humanas e computacionais. O homem possui um conjunto de faculdades que podem ser compartilhadas, direta ou indiretamente, com outros participantes, entre elas as faculdades físicas, cognitivas, sensoriais, emocionais, entre outras. As faculdades sensoriais são consideradas faculdades fundamentais em uma aplicação de Simbiose Digital, ou seja, toda aplicação de Simbiose Digital deve compartilhar pelo menos uma faculdade sensorial, preferencialmente a visão e audição. O computador também possui faculdades que podem ser compartilhadas. Essas faculdades tentam aproximar-se, em grande parte, de algumas faculdades humanas, principalmente as faculdades relacionadas à memória (recuperação e armazenamento de informações), cognição (inteligência artificial e redes neurais), raciocínio (execução de aplicações e tomada de decisões), faculdades sensoriais (visão computacional e reconhecimento de voz), entre outras. As faculdades computacionais possuem uma grande vantagem em relação às faculdades humanas quanto à capacidade e velocidade no processamento das informações, que são de grandeza superiores às faculdades humanas.

Uma faculdade por ser adquirida ou natural. O homem já nasce, normalmente, com o sentido da visão, que é uma faculdade natural, e ao longo da sua vida pode adquirir outras faculdades, algumas são adquiridas pelo próprio convívio na sociedade como falar e andar, outras são adquiridas intencionalmente como o conhecimento em alguma área da ciência. O computador naturalmente já possui a capacidade de processar e armazenar um grande conjunto de dados, e conforme as aplicações que ele executa ou os dados que ele manipula, ele adquire a capacidade de executar tarefas específicas.

As atuais aplicações colaborativas focam o compartilhamento de um mesmo ambiente entre os usuários, onde cada usuário possui a sua visão do ambiente, principalmente nas aplicações baseadas em sistemas de Realidade Virtual e Realidade Aumentada. Uma aplicação de Simbiose Digital visa compartilhar as faculdades humanas e computacionais. O compartilhamento do ambiente é uma conseqüência do compartilhamento das faculdades humanas (visão e audição). Compartilhar as faculdades humanas permite que as experiências vividas por um participante sejam compartilhadas com os demais elementos simbiontes, permitindo que todos os elementos participantes possam colaborar ou apenas vivenciar essas experiências. Os participantes de uma aplicação de Simbiose Digital (seres humanos e computadores), podem ser ativos ou passivos quanto a uma dada faculdade. Os ativos são aqueles que permitem que uma ou mais faculdades suas, sejam compartilhadas com os demais participantes. Já os passivos, são aqueles que tiram algum benefício das faculdades dos participantes ativos. A simbiose entre homens e computadores, através do compartilhamento das faculdades naturais e adquiridas, acrescenta um novo paradigma no modo como um usuário pode relacionar-se com um computador e com outros usuários ao mesmo tempo. Isso possibilita que os usuários participantes de uma aplicação de Simbiose Digital exerçam um ou mais papéis conforme mostra a Tabela 1. A representação dos papéis pode ser estática, mudar durante a aplicação ou ainda um papel ou mais papéis podem ser suprimidos conforme as características de cada aplicação.

3.3 Ambiente de Experimentação Sensorial Coletiva

O compartilhamento das faculdades sensoriais permite que a experimentação sensorial de um seja a experimentação sensorial de todos. Um ambiente de experimentação coletiva permite que dois ou mais usuários sejam submetidos às mesmas experiências sensoriais. Isso permite a construção coletiva do conhecimento e que as competências de todos os participantes em conjunto com os sistemas computacionais sejam agrupadas virtualmente na execução de alguma tarefa.

A consciência pode ser definida como o atributo pelo qual o ser humano pode conhecer e julgar sua própria realidade. O nível de consciência do ambiente de cada usuário está diretamente relacionado à quantidade de sentidos compartilhados e aos meios utilizados para permitir a experiência da presença. Um ambiente de experimentação coletiva é uma das principais características de uma aplicação de Simbiose Digital. Isso permite que todos os participantes tenham atenção somente em um único foco, e juntamente com os sistemas computacionais, proporcionam um ambiente com características únicas não contempladas totalmente em outros tipos de aplicações. Dessa forma, a consciência de cada um sobre o ambiente e sobre os eventos que ocorrem nele pode ser construída coletivamente.

Tabela 1 – Tipo de Papéis

Tipos de Papéis	Descrição
Aprendiz	Esse papel está relacionado aos usuários que participam do processo colaborativo com objetivos de aprendizagem ou treinamento.
Colaborador	Esse papel é voltado para usuários que participam diretamente de algum processo colaborativo.
Coordenador	Esse papel deve ser representado por usuários com papel de liderança durante o processo colaborativo para a tomada de decisões e atribuições de tarefas.
Facilitador	O papel do facilitador é manter o grupo de aprendizes interagindo em um processo colaborativo. Esse usuário deve agir como líder, mas não deve conduzir o processo colaborativo.
Observador	O usuário que exerce esse papel não interage com nenhum dos usuários da aplicação, a não ser com outros observadores. Ele pode ser apenas um observador ou ainda ser um avaliador para os demais usuários da aplicação ou do próprio sistema.
Professor	O professor, diferente do facilitador, conduz o processo colaborativo, além de manter a interação dos usuários.
Suporte	Esse usuário ajuda o professor ou facilitador no gerenciamento da aplicação, levantamento de informações adicionais, entre outras atividades. Ele pode ser ainda um usuário de área diferente do professor ou facilitador, ou ainda um psicólogo, terapeuta ocupacional, sociólogo, etc.

3.4 Ambiente Compartilhado e Localização dos Usuários

O compartilhamento do ambiente acontece pelo compartilhamento das faculdades humanas, mais especificamente dos sentidos humanos. Uma aplicação de Simbiose Digital pode ter mais que um ambiente compartilhado e eles podem ser ambientes puramente sintéticos (realidade virtual), ambientes puramente naturais (mundo real) ou ambientes mistos (realidade mista, realidade aumentada ou virtualidade aumentada).

Em uma aplicação de Simbiose Digital existem pelo menos dois usuários participantes. Uma aplicação de Simbiose Digital pode ser também uma aplicação de tele-presença, porém a diferença é que ao invés de existir um robô como elemento de atuação, existirá uma pessoa. Associações de dois ou mais usuários em um mesmo ambiente também podem acontecer, o que não caracteriza uma aplicação de tele-presença.

3.5 Requisitos de uma aplicação

Os requisitos de uma aplicação de Simbiose Digital permeiam em grande parte os requisitos das aplicações de RV, RA, Vídeo Conferência, Tele-Presença, Multimídia, RM - Realidade Mista, entre outras. O uso ou não de cada uma dessas tecnologias depende das características de cada aplicação, pois podem existir aplicações de Simbiose Digital puramente baseadas em elementos reais que usam somente meios de comunicação baseados em vídeo e áudio, ou aplicações puramente sintéticas, como as aplicações de RV. Porém o principal requisito de uma aplicação de Simbiose Digital é permitir que os simbiontes possam compartilhar das suas faculdades em tempo real de forma natural e transparente. O compartilhamento das faculdades humanas permite que todos os usuários da aplicação possam compartilhar as mesmas experiências sensoriais que um usuário que as esteja vivenciando em um ambiente real ou sintético, e nesse relacionamento todos podem colaborar, ter objetivos próprios ou ainda serem meros espectadores.

O compartilhamento das faculdades é baseado no uso de elementos multimídia como vídeo, áudio, texto, gráficos e imagens geradas em tempo real. Com isso é necessário existir um comprometimento entre o tempo de resposta e a qualidade da apresentação, mesmo quando o número de participantes aumenta. Como em um sistema de Realidade Virtual, as aplicações de Simbiose Digital deverão ser capazes de reagir às ações dos usuários, refletindo-as para todos os participantes da aplicação com latência mínima [22]. Outros requisitos das aplicações de Sistemas Distribuídos de Realidade Virtual, também podem ser herdados, como o tratamento de múltiplos dispositivos de E/S (capacetes, óculos, dispositivos de rastreamento de movimentos, dispositivos de tato de força, entre outros); suporte a diferentes tipos de unidades de dados, com requisitos diferentes de confiabilidade na entrega; taxas rápidas de atualização; atualização do ambiente a taxas que variam de 10 a 60 qps; entre outros. Os requisitos herdados das aplicações multimídia dependem também dos tipos e das características das mídias utilizadas para o compartilhamento das faculdades humanas. As mídias utilizadas podem ser baseadas ou não no tempo. As baseadas no tempo como um vídeo ou um áudio, devem tratar dos aspectos relacionados em como manter o relacionamento temporal e o paralelismo durante a reprodução. Do mesmo jeito, os requisitos para os demais tipos de aplicações como tele-presença, realidade aumentada, virtualidade aumentada, realidade mista, entre outros, podem também ser considerados nas aplicações de Simbiose Digital.

4. CONSTRUÇÃO DE UMA APLICAÇÃO DE SIMBIOSE DIGITAL.

Vários tipos de aplicações encontradas na literatura possuem algumas características relacionadas às aplicações de Simbiose Digital, como os propostos por Goldberg et al [7], Billinghurst e Kato [2] e Joseph [16]. Porém elas não contemplam totalmente todas as características definidas para uma aplicação de Simbiose Digital, ou não foram construídas com esse objetivo. Em linhas gerais, uma aplicação de Simbiose Digital disponibiliza um ambiente de experimentação sensorial coletiva com suporte à colaboração entre todos usuários em tempo real. Várias aplicações suportam colaboração, mas cada usuário possui a sua visão do ambiente, ou seja, não é um ambiente de experimentação sensorial coletiva. Algumas possuem características de experimentação coletiva, mas não tem o foco em colaboração como o Tele Ator [7]. Algumas ferramentas de colaboração podem ser classificadas como uma aplicação de Simbiose Digital,

como um editor colaborativo, porém o ambiente compartilhado fica limitado ao da ferramenta (geralmente 2D). A versão de uma aplicação de Simbiose Digital inicialmente desenvolvida [27], foi ampliada e será discutida a seguir [28].

4.1 Descrição Geral da Aplicação

A aplicação proposta visa implementar um ambiente de experimentação coletiva voltado para tele-visitas. Nessa aplicação, um usuário local compartilha o ambiente com os demais usuários através da visão, audição e fala com mobilidade através de uma rede sem fio. A visão do ambiente é única e todos os usuários podem conversar entre si. Itens, objetos e salas desse ambiente podem ser identificados através de um marcador baseado na biblioteca ARTag [6]. A inclusão desses marcadores permite adicionar funcionalidades importantes como o rastreamento do usuário local, desenhos de rotas e o acesso às informações específicas conforme as necessidades e o perfil de cada usuário ou grupo. Toda experiência é armazenada e pode ser recuperada pelo usuário em um outro momento. O estudo de caso define uma ferramenta simples de anotações de vídeo digital baseado no padrão MPEG-7. Uma anotação define um conjunto de comentários inseridos em regiões que são destacadas em um quadro de vídeo. As anotações serão baseadas em segmentos de textos e permitem a adição de semântica ao conteúdo de um vídeo. Todo o sistema de comunicação está baseado na biblioteca GLASS [8][9]. Nesse estudo de caso, existe somente um usuário local, os demais usuários são remotos ao ambiente compartilhado. Os sentidos compartilhados por esse usuário são a visão, audição e fala.

4.2 Arquitetura de Usuário Local e Remoto

O usuário local difere dos demais usuários, principalmente pela captura do vídeo e de não possuir um teclado para a entrada de dados. Os dispositivos utilizados pelos os usuários remotos são computadores convencionais e não serão discutidos, exceto pelo uso de um monitor estéreo ativo para visualizar as imagens em estéreo, porém esse equipamento não é obrigatório.

4.2.1 Hardware - Usuário Local

Todos os componentes de *hardware* usados são componentes de mercado o que contribui para o barateamento do sistema e a produção em escala. O usuário local possui duas câmeras *FireWire* IEEE 1394 para compartilhar a sua visão em estéreo com os demais usuários. Essas câmeras permitem captura imagens digitais na resolução de 640x480 pontos com taxa de aquisição de 30 quadros por segundos. As duas câmeras são conectadas em um computador portável (Pentium IV com 512 MB de memória RAM). Como a maior parte dos computadores portáteis possuem somente uma porta FireWire IEEE 1394 sem a alimentação, duas soluções foram necessárias para o funcionamento das duas câmeras, o uso de uma bateria externa de 12 volts para a alimentação e o uso de um HUB FireWire para aumentar o número de portas. Essas duas câmeras estão dispostas em um capacete (Figura 2), juntamente com um microfone para a captura da voz e do áudio ambiente. Finalmente, todas as informações são transmitida por uma rede sem fio IEEE 802.11g, e isso é feito através de um cartão PCMCIA DLink 802.11 a/b/g, conectado ao computador portável. Todos esses componentes são alocados em uma mochila que deve ser carregada pelo usuário local (Figura 3). O usuário possui um HMD I-Glasses PC 3D para visualizar as imagens capturadas pelas câmeras e as informações geradas pela aplicação. O HMD possui suporte para visão e áudio estéreo, e também é alimentado por uma bateria externa. Para interagir com a aplicação, o usuário possui ainda um mouse de mão sem fio.

4.2.2 Aquisição de Imagens Estereoscópicas

Baseado na disparidade binocular [19], as duas câmeras foram dispostas no capacete, de modo a garantirem uma paralaxe próxima a visão humana, em torno de 70 milímetros (Figura 2). Após a captura, as imagens devem ser tratadas devido à natureza das lentes em relação a dois aspectos: a correção da distorção radial e o ajuste da luminosidade [29]. Tsai [30] discute amplamente os aspectos relacionados à calibração da câmera e também a distorção radial. Para obter todos os parâmetros necessários para a calibração das câmeras, esse trabalho utilizou a ferramenta e o método de calibração de Bouguet [3]. Após o tratamento das imagens, elas podem ser utilizadas localmente e serem também enviadas para os demais usuários participantes, sendo necessário aplicar algum método de compressão. Jiang e Edirinsinghe [14] propuseram um método híbrido utilizando a codificação JPEG e extração dos contornos e *edge* de ambas imagens. Com isso uma baixa taxa de bits pode ser produzida (alta compressão), porém a um custo computacional proibitivo para codificação em tempo real. Uma outra abordagem é a técnica denominada "simulcast", que é baseada na codificação independente do olho direito e esquerdo, tendo como ponto negativo a necessidade da duplicação dos serviços de codificação e transmissão. Essa técnica é adotada nesse trabalho e também adotada pelo padrão MPEG-2, porém não é reconhecida como uma codificação eficiente [13]. Sentinens et al [24], demonstraram que a codificação JPEG não afeta a percepção de profundidade do par estéreo. Para alcançar os melhores resultados, entre a maior taxa de compactação e o menor tempo de codificação e adequação a largura de banda disponível para a transmissão das imagens, vários níveis de codificação em relação à qualidade de codificação que o JPEG foram testados através da biblioteca libjpeg. Os níveis de qualidade testados foram de 100%, 80%, 60%, 40% e 20% de qualidade. Com as taxas de 20% e 40%, a percepção da profundidade foi boa, porém a experiência do usuário em relação à qualidade foi problemática, e a taxa em quadros por segundo codificadas ficaram em torno de 25. Com o parâmetro de qualidade de codificação entre 60% e 80%, a experiência do usuário em relação a qualidade melhorou, com a percepção de profundidade muito boa e as taxa em quadros por segundo ficaram em torno de 20. Dois modos de visualização são definidos pela aplicação: tela cheia e normal. O usuário local deve operar, preferencialmente, no modo tela cheia, porém o modo normal oferece opções extras que serão discutidas nos próximos tópicos. Um menu está disponível em ambos os modos para a seleção de algumas opções da aplicação que também serão discutidas nos próximos tópicos. O menu apresentado é um menu de "rolagem", onde o usuário pode trocar (rolar) os itens do menu através das duas setas apresentadas do lado esquerdo do menu (setas acima e abaixo. Veja Figura 4 e Figura 5).

4.2.3 Aquisição e Reprodução de Áudio

Uma solução para tornar a aplicação independente da plataforma foi a adoção da biblioteca OpenAL – Open Audio Library . Na aplicação proposta, o áudio capturado possui taxa de amostragem de 8000Hz, canal mono e quantização de 8 bits. Como a taxa de bits gerada fica em torno de 7Kbytes por segundo, nenhum tipo de codificação (compactação) foi aplicado para diminuir o tamanho

de banda ocupada pelo áudio durante a transmissão. Apesar de ser uma qualidade baixa para a reprodução do áudio nos clientes remotos, não existe nenhuma perda do significado da voz durante a reprodução, somente alguns eventos de áudio oriundo do ambiente local ficam prejudicados em relação à qualidade durante a reprodução. Para o controle a interação por voz, o usuário local (guia) pode ativar a opção de todos os usuários remotos falarem ao mesmo tempo ou somente um usuário remoto falar por vez para evitar confusão. Se a opção de um usuário remoto falar por vez estiver ativa, um usuário remoto para falar deve primeiro ter o controle do microfone. Se o microfone estiver disponível, o usuário remoto pode comunicar-se com os demais, caso contrário, ele entra na fila de espera. Ao final, o usuário remoto deve liberar o microfone. Para facilitar essa operação, a aplicação indica se o microfone está disponível ou não e quantos usuários estão na fila de espera. A aplicação também mostra a imagem do usuário que está falando (Figura 4).

4.2.4 Reconhecimento de Padrões

O reconhecimento de padrões ou marcadores é muito utilizado em aplicações de realidade aumentada. A biblioteca ARTag [6] é inspirada na biblioteca ARToolKit [2], que foi desenvolvida para dar suporte ao desenvolvimento de aplicações de realidade aumentada. A ARTag possui algumas funções que melhoraram o reconhecimento dos marcadores aumentando a taxa de acerto em relação a distância e o ângulo da câmera em relação ao marcador, em comparação a biblioteca ARToolKit [2], porém a biblioteca ARTag não possui o código fonte aberto e não pode ser usada para fins comerciais sem a autorização dos autores. Na implementação proposta, a aplicação ao reconhecer um marcador, apresenta no canto superior esquerdo a letra "I" de Informação. Isso indica para o usuário que informações extras estão disponíveis como imagens, texto, animação e endereços na WEB. A aplicação WEB recebe o número do marcador e o ID do usuário, e pode realizar qualquer operação com essas informações. Como a aplicação WEB é independente da aplicação gráfica, ela pode ser customizada de várias formas. A letra "I" possui dois estados indicados pela cor. Verde significa que o marcador foi reconhecido a menos de 15 segundos. Vermelho indica que o marcador foi reconhecido a mais de 15 segundos. Após 40 segundos no estado vermelho, ela desaparece. Dessa forma, cada usuário poderá selecionar e visualizar em detalhes, as informações disponíveis dentro do intervalo estipulado. A visualização dessas informações é feita de forma independente por cada usuário, ou seja, cada usuário poderá interagir e visualizar essas informações, sem afetar os demais usuários da aplicação. Essas informações são replicadas para todos os usuários e são organizadas em um sistema de arquivos. Um outro aspecto importante é que as opções do usuário administrador podem ser diferentes dos demais usuários, como por exemplo, qual roteiro ele deve seguir pelo museu, dicas sobre as informações de cada obra, entre outras.

4.2.5 Rastreamento do Usuário Local

Devido às características do estudo de caso proposto, as formas de rastreamento tradicionais [25] não são adotadas nesse projeto, pois além dos fatores de custo financeiro e computacional, o rastreamento do usuário não precisa ser exato. Para isso, os marcadores também são utilizados para rastrear o usuário local. Isso permite que todos os participantes possam saber, por exemplo, em que sala o usuário local está. Para que isso seja permitido, cada item ou objeto, devem possuir um ou mais marcadores. A principal vantagem dessa abordagem é em relação ao custo financeiro, já que não necessita de nenhum equipamento especial, só um dispositivo de captura de imagens já utilizado para a captura das imagens do ambiente local. Nesse tipo de abordagem, o rastreamento do usuário local é em relação ao último marcador identificado. Como isso pode causar possíveis erros de localização, é informado também o tempo, em segundos, desde o último reconhecimento. Caso o ponto esteja verde, isso significa que a localização do usuário é precisa. O ponto vermelho indica o último marcador reconhecido. Todos os usuários podem realizar operações de *zoom-in, zomm-out*, translação e rotação no mapa. Todas as coordenadas referentes ao mapa, marcadores e informações estão armazenadas em um arquivo de texto e dessa forma, a alteração do mesmo torna-se simples. Uma outra característica implementada é a busca pelo menor caminho entre a sala atual e uma sala ou objeto desejado. Por se tratar de um problema clássico na literatura, o algoritmo de Dijkstra [23] foi escolhido para implementar essa funcionalidade. Um grafo com uma certa quantidade de vértices e o custo entre eles são definidos, conforme o posicionamento dos marcadores pelas salas. Dado o vértice inicial e final, o algoritmo identifica o caminho com o menor custo dentre os vértices existentes.

4.2.6 Anotação de Imagens

Descrever um conteúdo multimídia permite a construção de aplicações voltadas para busca, consulta, filtragem e navegação desse mesmo conteúdo. As informações utilizadas para fazer a indexação podem ser obtidas, por exemplo, a partir das características visuais (movimentação, cor, intensidade, entre outras) ou do áudio associado ao vídeo. Uma outra forma de obter essas informações para fazer a indexação, é a partir da semântica de seu conteúdo representada sob a forma de anotações textuais [17]. A forma mais eficiente para obter a semântica de um conteúdo visual é através das anotações textuais, pois a extração automática da semântica é um problema que continua em aberto. As anotações textuais são transformadas e estruturadas em metadados, e associadas ao conteúdo visual. Vários modelos de metadados são encontrados na literatura, como RDF [4] e o MPEG-7 [20], sendo o padrão MPEG-7 o utilizado nesse trabalho. Um simples sistema de anotação colaborativa de imagens foi implementado baseado no uso do padrão MPEG-7. O sistema implementado visa somente gerar os metadados e o armazenamento das experiências colaborativas (vídeo, imagens e áudio). A biblioteca MPEG-7 Library2, foi utilizada para facilitar a manipulação dos descritores pela aplicação. Ela é gratuita e é escrita em C++. Na aplicação proposta, as anotações são feitas somente sobre uma imagem estática. O usuário local, ao acionar a opção de anotação, irá congelar o vídeo de todos os participantes. A anotação é feita sobre uma área retangular definida pelo usuário e não é permitido realizar anotações simultâneas. Como o usuário local não possui meios eficientes (teclado) para realizar as anotações, essa tarefa é destinada somente aos usuários remotos. As anotações são associadas ao vídeo tomando como referência o número do quadro e o tempo. Ao finalizar uma anotação, um arquivo MPEG-7 no formato textual é gerado pela aplicação. Existem alguns tipos de informações que podem ser geradas

[2] http://iiss039.joanneum.at/cms/index.php?id=84

automaticamente pela aplicação, como data, hora, local, participantes, entre outras. Porém, informações mais específicas podem ser geradas automaticamente pela aplicação quando um marcador for reconhecido. Isso permite adicionar conteúdos semânticos significantes de forma automática, conforme o relacionamento de cada marcador com o mundo real. O trecho de áudio compreendido entre o início e fim da anotação é associado a essa anotação automaticamente. (Figura 5).

5. CONCLUSÕES

5.1 Considerações sobre Simbiose Digital

Nos vários tipos de aplicações computacionais que possibilitam a colaboração baseadas em ambientes 3D, cada usuário geralmente possui o seu ponto de vista e o poder de decidir o que fazer, ou seja, a experiência da presença deste usuário nesses ambientes espelha-se no mundo real, sendo isso muitas vezes desejado. Nessas aplicações o compartilhamento do ambientes é o foco principal, mas com estreitamento do relacionamento entre homens, computadores e os meios de comunicação, um novo paradigma de colaboração pode ser explorado baseado no compartilhamento das faculdades humanas e computacionais, que nesse artigo é denominado de Simbiose Digital. Possibilitar a experiência de presença através do compartilhamento das faculdades humanas e computacionais permite explorar um novo universo de aplicações que podem expandir as fronteiras da colaboração entre homens e computadores. Aplicações com foco em experimentação coletiva com suporte a colaboração, permitem maximizar as competências de cada usuário e explorar novas formas de cognição. As atuais aplicações não focam nos aspectos relacionados ao compartilhamento das faculdades humanas e computacionais em uma mesma aplicação ou ainda aspectos relacionados à experimentação coletiva. Essa mudança de paradigma do compartilhamento de um ambiente para o compartilhamento das faculdades humanas e computacionais é a principal diferença das atuais aplicações para as aplicações de Simbiose Digital.

Uma aplicação de Simbiose Digital pode ser classificada também como uma aplicação colaborativa. Uma aplicação de Simbiose Digital pode possuir as mesmas características das aplicações citadas por Benford et al [1] e em todas essas aplicações (Espaços de Mídia, Vídeo Conferência Espacial, Ambiente Virtual Colaborativo, Tele-Presença e Ambiente de Realidade Aumentada Colaborativo), cada usuário age de forma independente, imitando o mundo real. No geral, uma aplicação de Simbiose Digital é mais indicada quando o objetivo for concentrar todas as atenções e energias dos participantes em um mesmo foco, através do compartilhamento das faculdades humanas e computacionais. Uma aplicação de Simbiose Digital pode também agregar as características de uma aplicação de tele-presença, porém usa o ser humano ao invés de um robô como agente de interação com o meio ambiente compartilhado, permitindo uma interação mais amigável entre os participantes remotos e locais.

O uso da visão, tato, áudio e elementos gráficos não são exclusivos das aplicações de Simbiose Digital, porém aqui todas essas características podem ser reunidas em uma mesma aplicação e serem usadas de formas diferentes, pois têm o objetivo de compartilhar as faculdades humanas e computacionais. Compartilhar um ambiente através das faculdades sensoriais de uma pessoa permite que um ou mais usuários remotos sejam expostos às mesmas experiências sensoriais, e isso não é o foco das atuais aplicações de CSCW, porém todas elas partilham de um mesmo meio de comunicação e interação baseados em elementos multimídia. As aplicações de CSCW podem aplicar um conceito conhecido por visão convergente ou WYSIWIS (*What You See Is What I See* – O que você vê é o que eu vejo). Isso está próximo ao conceito de experimentação sensorial coletiva quanto ao sentido da visão discutido nesse trabalho, mas são conceitos distintos. A visão convergente é uma abstração para a construção de interfaces gráficas que permitam simular as mesmas características de um encontro face a face em frente a uma lousa [26]. O conceito de experimentação sensorial coletiva é mais amplo. Fazendo uma analogia com o conceito WYSIWIS, a experimentação sensorial coletiva pode ser definida como: A sua experiência sensorial é a minha experiência sensorial. As atuais taxonomias das aplicações colaborativas não mencionam uma classificação baseada no compartilhamento das faculdades humanas ou ainda em ambientes de experimentação coletiva como em [12][15][31]. Um novo elemento pode ser adicionado nas taxonomias das aplicações colaborativas quanto à experiência da presença, que pode ser individual ou coletiva. As aplicações de Simbiose Digital podem englobar todos aspectos relacionados às cinco categorias descritas por Benford et al [1] e variar, tanto nos aspectos relacionados à artificialidade como no de transporte.

Uma aplicação de Simbiose Digital pode possuir também as características das aplicações de Realidade Mista, porém podem existir aplicações de Simbiose Digital baseadas somente em uma das extremidades do espectro definido por Milgram et al [2], usando somente elementos reais ou virtuais dentro do conceito de realidade e virtualidade continuada, ou seja, não é necessário existir uma fusão entre o sintético e o real, mas essa fusão é bem vinda e agrega novos benefícios e possibilidades nas aplicações de Simbiose Digital. Um outro fato é que em uma aplicação de Simbiose Digital pode existir uma fusão de ambientes em uma mesma aplicação.

A aplicação desenvolvida por Goldberg et al [7] pode ser considerada um caso particular de uma aplicação de Simbiose Digital por compartilhar a visão de um usuário com os demais participantes. A existência do diretor é um elemento de vital importância para a aplicação, pois ele sugere os próximos passos para os usuários remotos e para o tele ator, ou seja, o tele ator não tem um contato direto com os usuários remotos, fazendo apenas o papel de agente que atua localmente na aplicação, pois ele é dirigido pelo diretor. Um outro aspecto é que não existe nenhum tipo de comunicação entre os participantes remotos. Com isso, a interação entre os participantes e o agente local é direcionada, o que restringe a liberdade de criação e colaboração dos usuários. Goldberg et al [7] caracterizam a sua aplicação dentro da taxonomia proposta por Chong et al [5] para sistemas tele operados que se dividem em 4 grupos, baseados na quantidade de operadores e robôs utilizados na aplicação: um para um, vários para um, vários para vários e um para vários.

Dentre todas as aplicações consideradas, nenhuma trata dos aspectos relacionados ao compartilhamento das faculdades humanas e computacionais em uma mesma aplicação, ou ainda, aspectos relacionados à experimentação sensorial coletiva. Essa mudança de foco do compartilhamento de um ambiente para o compartilhamento das faculdades humanas e computacionais é a principal diferença das atuais aplicações para as aplicações de Simbiose Digital. A experiência de presença através do compartilhamento das faculdades humanas e computacionais

permite explorar um novo universo de aplicações que pode expandir as fronteiras do uso da informática em diversos tipos de aplicações.

5.2 Considerações sobre a aplicação proposta

A aplicação foi testada em ambiente de laboratório com cinco usuários, mas está pronta para ser executada em um ambiente real e pode ser usada, por exemplo, em ambientes de tele-visita a museus. A aplicação suporta o aumento do número de usuários, porém o sistema de comunicação por voz pode se tornar crítico conforme a quantidade de usuários participantes aumenta. Alguns problemas foram identificados. O movimento da cabeça do usuário local pode ser critico caso ele mova a cabeça ou ande de forma rápida e constante. O peso total do equipamento do usuário local, em torno de 5 kg, pode causar desconforto para experiência de longa duração, mas isso deve ser relevado por se tratar de um protótipo. O tempo de vida das baterias também é uma preocupação. Atualmente o sistema pode trabalhar por um período de até 2 horas, mas isso pode ser melhorado através do uso de baterias melhores (maior tempo de vida). A qualidade do áudio capturado também pode ser crítica conforme o tipo de experiência desejada. Apesar de ser uma qualidade baixa para a reprodução do áudio, não existe perda do significado da voz durante a reprodução e somente alguns eventos de áudio oriundo do ambiente local ficam prejudicados em relação à qualidade durante a reprodução. A poluição visual causada pelos marcadores pode ser inadequada em alguns tipos de ambientes, mas os marcadores oferecem uma solução simples para o rastreamento do usuário e obtenção de informações específicas sobre objetos de um ambiente. O sistema de rastreamento fornece somente a informação do posicionamento do usuário local, mas com o uso de marcadores no teto, a orientação do usuário também pode ser obtida. A ferramenta de anotação textual é simples, mas define um modelo de anotação colaborativa em tempo real. A área de cobertura do Ponto de Acesso à rede sem fio limita a mobilidade do usuário local. Os protocolos WLAN (*Wireless Local Area NetWork*) utilizados podem transmitir o vídeo dentro das necessidades da aplicação entre 22 e 25 quadros por segundo. A largura de banda e a taxa de quadros por segundo podem ser melhoradas com a substituição da tecnologia WLAN.

6. REFERÊNCIAS

[1] Benford, S. et al. Understanding and Constructing Shared Spaces with Mixed Reality Boundaries. ACM Transactions on Computer-Human Interaction (TOCHI) archive Volume 5, Issue 3. p.185-223, 1988.

[2] Billinghurst, M. e Kato, H. "Collaborative Mixed Reality". In Proceedings of the First International Symposium on Mixed Reality (ISMR'99). Merging Real and Virtual Worlds. Berlin: Spring Verlag. 1999, p. 261-284.

[3] Bouguet, J.-Y. Camera Calibration Toolbox for Matlab, 2004. http://www.vision.caltech.edu/bouguetj/calib doc/ (Acesso em Julho 2004).

[4] Cherry, S.M. Weaving a Web of ideas IEEE Volume 39, Issue 9, Sep 2002 Page(s):65 – 69.

[5] Chong, N. et al. Remote coordinated controls in multiple tele-robot cooperation. In: IEEE International Conference on Robotics and Automation. p.3138-3343, 2000.

[6] Fiala, M. ARTag Revision 1, A Fiducial Marker System Using Digital Techniques. Tecnical Report. National Researh Council Canada. 2004.

[7] Goldberg, K., Song, D., Khor, Y., Pescovitz, D., evandowski, A., Himmelstein, J., Shih, J., Ho, A., Paulos, E. e Donath, J. "Collaborative Online Teleoperation with Spatial Dynamic Voting and a Human: Tele-Actor". IEEE International Conference on Robotics and Automation, May 2002. p. 1179-1184.

[8] Guimarães, M. et al. Ambientes Virtuais: Projeto e Implementação. In: Alexandre Cardoso; Cesar Augusto Camilo Teixeira e Edgard Lamounier Júnior. (Org.). Porto Alegre. p.114-133, 2003.Suppressed to the blind review.

[9] Guimarães, M *Powering Multiprojection Immersive Environments with Cluster of Commodity Computers* . SIACG, 2002..

[10] Happer, D. Online Etymology Dictionary. 2001. http://www.etymonline.com/

[11] Houaiss, A. V. e Villas, M. S. Dicionário Houaiss da Língua Portuguesa. Objetiva, 2001. 2922 pags.

[12] Jarczyk, A., Löffler, P., Völksen, G.: "Computer Supported Cooperative Work (CSCW) State of the Art". Version 1.0, Internal Report, Siemens AG, Munich, 1992.

[13] Javidi, B. and OKANO, F., Three-Dimensional Television, Video, and Display Technologies, Springer, 2002

[14] Jiangm J. and EDIRISINGHE, E.A., A hybrid scheme for low bit-rate coding of stereo images, IEEE Transactions on Image Processing, Volume 11, Issue 2, pp. 123-134, February 2002

[15] Johansen, R. Groupware: Computer support for business teams. New York: The Free Press, 1988.

[16] Joseph, J. K. "Making Scents: aromatic output for HCI". IEEE Interactions: Volume 11, Issue 1, January + February 2004, p.48-61

[17] Kokkoras, F., Jiang, H., Vlahavas, I., Elmagarmid, A. K., Houstis, E. N. and AREF, W. G. Smart VídeoText: A Video data model base don conceptual graphs. Multimídia Systems, July, 2002. p.328 - 338.

[18] Licklider, J. C. R. (1960) "Man-Computer Symbiosis", RE Transactions on Human Factors in Electronics: volume HFE-1, March 1960. p.4–11.

[19] Lipton, L. Stereo3D Handbook. Stereographics Corporation. 1997. Disponível online em: http://www.stereographics.com

[20] Manjunath, B. S., Salembier, P; Sikora, T. Introduction to MPEG-7. Editora: John Wiley & Sons, LTD. 1ºedição. 2002'.

[21] Milgram, P. et al F. Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. In: SPIE Proceedings: Telemanipulator and Telepresence Technologies . H. Das, SPIE. 2351, p.282-292, 1994.

[22] Pullen, J. M. and Laviano, V. P. A. A Selectively Reliable Transport Protocol for DIS. 13rd Workshop on Standards for Interoperability of DIS. Florida: 1995.

[23] Sedgewick, R. Published by Addison Wesley Professional. Algorithms in C++ Part 5: Graph Algorithms, 3rd Edition, 2001. pág 528.

[24] Seuntiens, P., Lydia Meesters, and Wijnand A. IJsselsteijn, Perceptual evaluation of JPEG-coded stereoscopic images, Stereoscopic Displays and Virtual Reality Systems X, Proceedings of SPIE, volume 5006, pp. 215-226, May, 2003

[25] Silva, R. J. M., "Integração de um Dispositivo Óptico de Rastreamento a uma Ferramenta de Realidade Virtual", pp. 18-24, novembro/2005.

[26] Stefik, M, Bobrow, D. G., Lanning, S., Tatar, D. and Foster, G. "WYSIWIS Revised: Early Experiences With Multi-User Interfaces". In Conference on Computer-Supported Cooperative Work. Texas, 1986. p276-290.

[27] Todesco, G. e Zuffo, M. K. Simbiose Virtual, In: Simposio de Realidade Virtual. Ribeirão Preto. VI SVR2003. Ribeirão Preto: SBC p.398-400, 2003.

[28] Todesco, G.; Simbiose Digital. Tese (Doutorado). Escola Politécnica, Universidade de São Paulo. São Paulo 2006. Simposio de Realidade Virtual. Ribeirão Preto. VI SVR2003. Ribeirão Preto: SBC p.398-400, 2006.

[29] Trucco, E. and VERRI, A. Intoductory Techniques for 3-D Computer Vision, Prentice-Hall, 1998.

[30] Tsai, R.Y. (1987). A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, pp. 323-344

[31] Völksen, G.: "Approach Strategies to Groupware". Proceedings Groupware `93 Europe Conferences in Stockholm, London, Frankfurt, The Conference Group, Scottsdale, AZ, 1993.

[32] Zuffo, M. K. A Convergência da Internet e da Realidade Virtual em Novos Paradigmas de TV Interativa. Tese (Livre Docência). 2001. Escola Politécnica, Universidade de São Paulo. São Paulo 2001.

Figura 2 – Câmeras montadas no capacete + HMD

Figura 3 –Todos os componentes fora da mochila

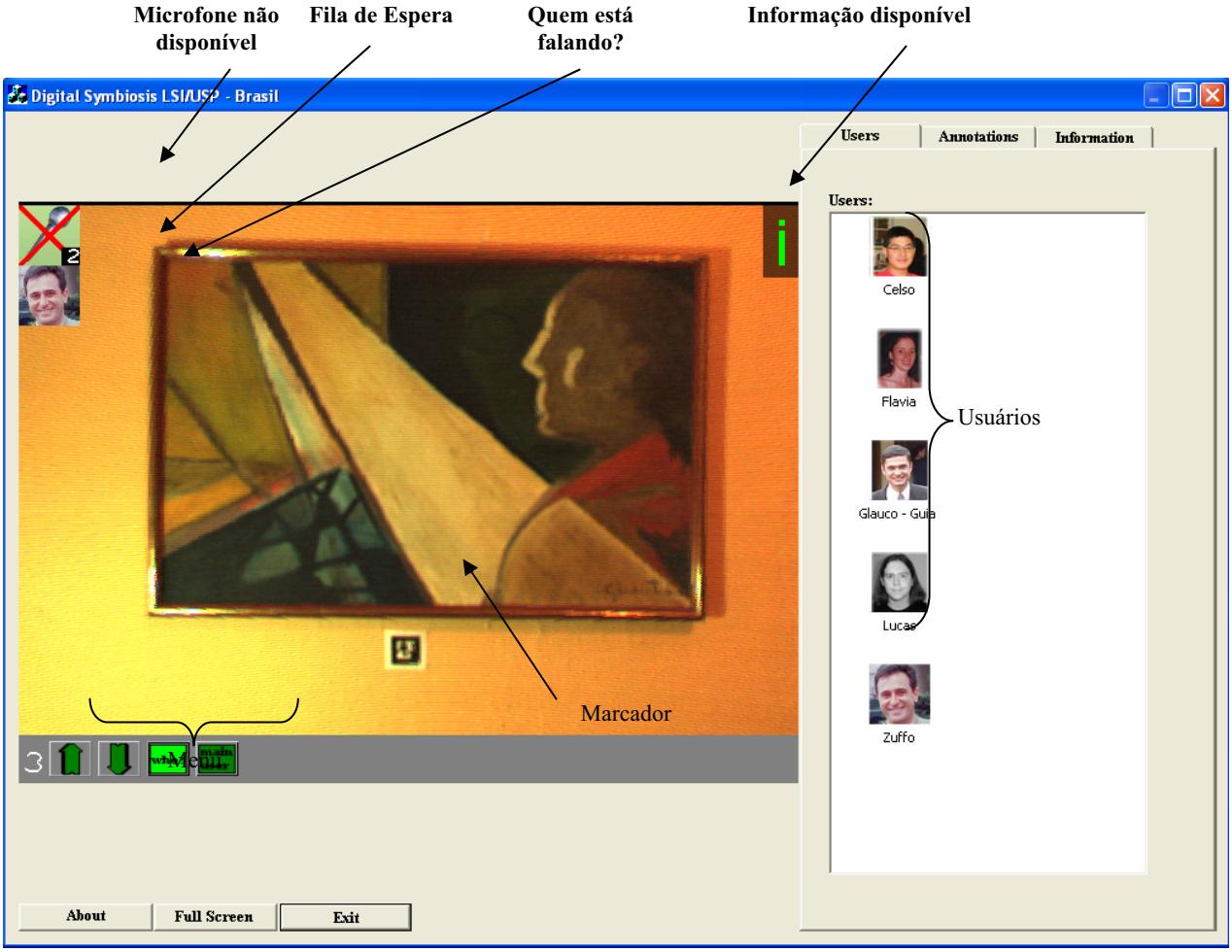


Figura 4 – Tela da Aplicação – Usuários participantes (Lado Esquerdo) e Disponibilidade do Microfone (canto superior direito)

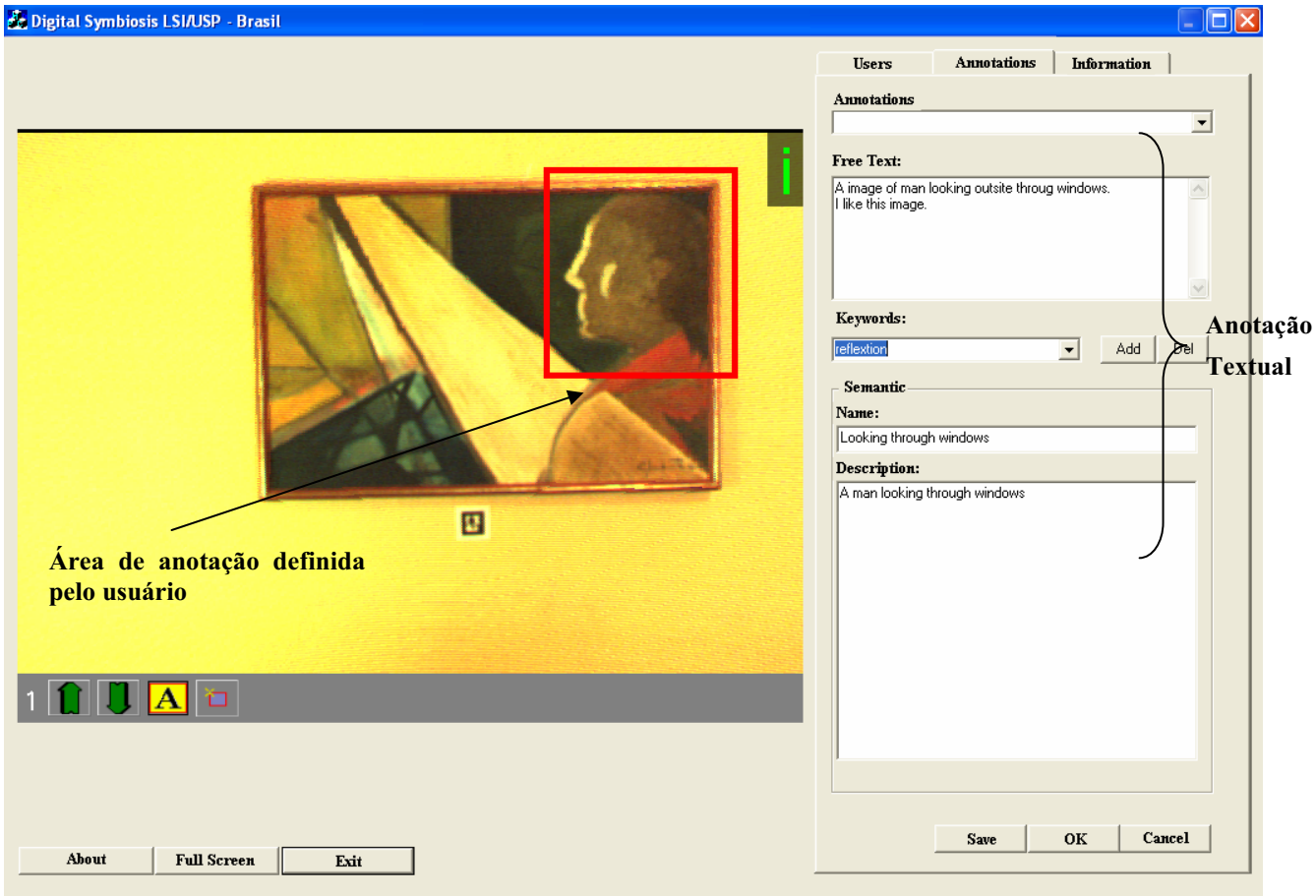


Figura 5 – Tela da Aplicação - Sistema de Anotação Textual (Modo Normal)