

**Anais do**  
**IV Simpósio Brasileiro de Sistemas Colaborativos**  
**SBSC 2007**

**Rio de Janeiro, RJ - Brasil**

**03 a 05 de Julho de 2007**

**Promoção**

Sociedade Brasileira de Computação  
Comissão Especial de Sistemas Colaborativos

**Realização**

Universidade Federal de Pernambuco (UFPE)  
Centro Universitário Vila Velha (UVV)  
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)  
Instituto Militar de Engenharia (IME)

**Edição**

Ana Carolina Salgado (UFPE)  
Marco Aurélio Gerosa (UVV)

© Sociedade Brasileira de Computação  
Anais do IV Simpósio Brasileiro de Sistemas Colaborativos – SBSC 2007

2007 – Direitos desta edição reservados à Sociedade Brasileira de Computação

ISBN xx-yyyyy-zz-x

**Editores**

Ana Carolina Salgado

Marco Aurélio Gerosa

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>

Catálogo na publicação elaborada pela Biblioteca Central /UVV-ES

S612a Simpósio Brasileiro de Sistemas Colaborativos (4. : 2007 : Rio de Janeiro, RJ)  
Anais do IV Simpósio Brasileiro de Sistemas Colaborativos, 03 a 05 de julho de 2007 / editores Ana Carolina Salgado, Marco Aurélio Gerosa. – Porto Alegre : Sociedade Brasileira de Computação, 2007.  
viii, 150 p.  
Inclui bibliografias.  
ISBN XX-XXXX-XXX-X

1. Computação - Congressos. 2. Grupos de trabalho - Processamento de dados - Congressos. I. Salgado, Ana Carolina. II. Gerosa, Marco Aurélio. III. Sociedade Brasileira de Computação. IV. Título.

CDD 004.06

## Apresentação

---

Sistemas colaborativos estão cada vez mais presentes na sociedade. Além das já tradicionais ferramentas de comunicação, como correio eletrônico, mensageiro, bate-papo, fórum e áudio e videoconferência, estão cada vez mais presentes sistemas de realidade virtual, blogs, editores cooperativos, redes peer-to-peer, sites de relacionamento, entre outros. Mesmo sistemas tradicionalmente monousuário passaram a incorporar funcionalidades voltadas para a colaboração.

O IV Simpósio Brasileiro de Sistemas Colaborativos (SBSC 2007) é mais um passo rumo à consolidação da comunidade nacional de pesquisa sobre a área. A área é inerentemente multidisciplinar, envolvendo engenharia de software, banco de dados, inteligência artificial, sistemas de informação, interação humano-computador, computação gráfica, sistemas multimídia, sistemas distribuídos, integração entre hardware e software, além de outras áreas como sociologia, psicologia e administração. Sistemas Colaborativos são aplicados em diversas áreas, como por exemplo, trabalho, educação e entretenimento.

Esta edição do evento é marcada pela realização em conjunto com o Congresso da Sociedade Brasileira de Computação (CSBC) e a conseqüente antecipação do evento que vinha sendo realizado no final do segundo semestre. O SBSC 2006 foi realizado em Natal (RN), em conjunto com o WebMedia e o IHC. Antes da criação e aprovação pela SBC da CESC (Comissão Especial de Sistemas Colaborativos), que aconteceu no final de 2005, o evento era denominado WCSCW (Workshop Brasileiro de Tecnologias para Colaboração). O WCSCW 2005 foi realizado em conjunto com o SBIE, em Juiz de Fora (MG), e o WCSCW 2004 foi realizado em conjunto com o WebMedia em Ribeirão Preto (SP). O evento originou-se da Trilha de CSCW (*Computer-Supported Cooperative Work*) no WebMedia 2003, em Salvador (BA).

O SBSC 2007 teve 23 submissões, cada uma avaliada por 3 revisores. Os 11 artigos mais bem avaliados foram aceitos no simpósio. Os autores dos melhores artigos foram convidados a submeter uma versão estendida em inglês para a revista *Scientia*.

Agradecemos ao comitê de programa do evento e aos demais revisores pelo belo trabalho, concluído no prazo estabelecido. Nossos especiais agradecimentos aos Profs. Flavia Santoro e Mariano Pimentel, coordenadores da organização do evento e ao Prof. Jacques Wainer, coordenador da CESC.

Desejamos que o SBSC 2007 seja um proveitoso fórum de debates e reflexão sobre os temas mais importantes na área de Sistemas Colaborativos.

Rio de Janeiro, julho de 2007.

Ana Carolina Salgado  
Marco Aurélio Gerosa  
Coordenadores do Comitê de Programa SBSC 2007

**Coordenação Geral:**

Flávia Maria Santoro (UNIRIO)  
Mariano Pimentel (UNIRIO)

**Coordenação do Comitê de Programa:**

Ana Carolina Salgado (UFPE)  
Marco Aurélio Gerosa (UUV)

**Comitê de Programa:**

Alberto Raposo (PUC-Rio)  
Ana Carolina Salgado (UFPE)  
André Luís Vasconcelos Coelho (UNIFOR)  
Autran Macêdo (UFU)  
Carlos André Guimarães Ferraz (UFPE)  
Carlos José Pereira de Lucena (PUC-Rio)  
Celso Massaki Hirata (ITA)  
Cesar Augusto Tacla (UTFPR)  
Cláudia Lage Rebello da Motta (UFRJ)  
Cleudson de Souza (UFPA)  
Cléver Farias (FFCLRP/USP)  
Clovis Torres Fernandes (ITA)  
Flávia Santoro (UNIRIO)  
Geraldo Bonorino Xexéo (UFRJ)  
Hugo Fuks (PUC-Rio)  
Ivan Luiz Marques Ricarte (UNICAMP)  
Jacques Wainer (UNICAMP)  
Jaime Simão Sichman (USP)  
Jano de Souza (UFRJ)  
Jauvane Cavalcante de Oliveira (LNCC)  
José Maria Nazar David (FRB)  
José Valdeni de Lima (UFRGS)  
Marco Aurélio Gerosa (UUV)  
Marco Aurélio Souza Mangan (PUCRS)  
Marcos Borges (UFRJ)  
Mariano Pimentel (UNIRIO)  
Neide dos Santos (UERJ)  
Patrícia Tedesco (UFPE)  
Renata Araujo (UNIRIO)  
Roberta Lima Gomes (UFES)

## Avaliadores

---

Alberto Raposo  
Amaury André  
Anarosa Brandão  
André Coelho  
Autran Macêdo  
Carlos André Guimarães Ferraz  
Celso Hirata  
Cesar Tacla  
Claudia Motta  
Cleidson de Souza  
Clever Farias  
Clovis Fernandes  
Eveline Sá  
Flavia Santoro  
Geraldo Xexeo  
Hugo Fuks  
Ivan Ricarte

Jacques Wainer  
Jano Souza  
Jauvane Oliveira  
José David  
José Valdeni de Lima  
Marco Mangan  
Marco Aurélio Gerosa  
Marcos Borges  
Mariano Pimentel  
Neide Santos  
Patrícia Tedesco  
Paulo Castro  
Reinaldo Bianchi  
Renata Araujo  
Roberta Gomes  
Sandra Siebra  
Tiago Telecken

**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)

**Comitê Gestor (Gestão 2007)**

**Membros:**

Jacques Wainer (UNICAMP) – Coordenador

Alberto Raposo (PUC-Rio)

Ana Carolina Salgado (UFPE)

Marco Aurélio Gerosa (UVV)

Cléver Farias (FFCLRP/USP)

Flávia Santoro (UNIRIO)

Hugo Fuks (PUC-Rio)

## Sumário

---

<b>Coerência de Cache e Percepção em Ambientes de Cooperação Móvel</b> Carla Diacui Medeiros Berkenbrock, Celso Massaki Hirata	1
<b>Utilização de RPG e MABS no Desenvolvimento de Sistemas de Apoio à Decisão em Grupos</b> Diana F. Adamatti, Jaime S. Sichman, Helder Coelho	15
<b>WebBEMS: um sistema baseado em componentes para o suporte à colaboração via Web</b> Cléver R. Guareis de Farias, Carlos E. Gonçalves, Marta Costa Rosatelli	29
<b>Uma comparação técnica das abordagens existentes para implementação de requisitos colaborativos em aplicações não colaborativas</b> Mauro C. Pichiliani, Celso M. Hirata	43
<b>Inter- e Intra-relações entre Comunicação, Coordenação e Cooperação</b> Hugo Fuks, Alberto Raposo, Marco A. Gerosa, Mariano Pimentel, Denise Filippo, Carlos J. P. Lucena	57
<b>Um Modelo para Gestão de Conhecimento Baseado em Contexto</b> Vanessa Tavares Nunes, Flavia Maria Santoro, Marcos R.S. Borges	69
<b>Sistema de Co-Navegação com Suporte a Áudio-Conferência</b> Christopher V. Lima, Roberto Willrich, Roberta Lima-Gomes, Guillermo de Jesús Hoyos-Rivera	83
<b>Modelo de Design Rationale para atendimento médico ambulatorial</b> Cleo Billa, Cláudia Barsottini, Jacques Wainer	97
<b>Giga-Colab: Uma Arquitetura de Ambiente Virtual Colaborativo com suporte a Serviços Multimídia</b> Raoni Kulesza, Luiz G. P. Alves, Fernando L. Almeida, Marcos F. Jardini, Marco A. V. M. Vasconcelos, Guido L. S. Filho, Regina M. Silveira, Graça Bressan	111
<b>Uma Infra-estrutura Colaborativa de Suporte ao Desenvolvimento Distribuído Baseado em Componentes</b> João Paulo F. de Oliveira, Talles Brito, Yuri Morais, Amílcar Soares, Adriana E. de Oliveira, Sebastião Rabelo Jr, Glêdson Elias	125
<b>Mo Porã Um sistema gerenciador de repositórios distribuídos e colaborativos no ambiente científico da Amazônia</b> Kleberon. J. Amaral Serique, Laurindo Campos dos Santos, Felipe Santos Costa & Jair M. F. Maia	139

# Coerência de Cache e Percepção em Ambientes de Cooperação Móvel

Carla Diacui Medeiros Berkenbrock<sup>1</sup>, Celso Massaki Hirata<sup>1</sup>

<sup>1</sup>Divisão de Ciência da Computação  
Instituto Tecnológico de Aeronáutica - ITA  
12228-900 - São José dos Campos - SP - Brasil

{diacui, hirata}@ita.br

**Abstract.** *The advance of mobile computing and the new ways of connectivity allow the integration of mobile devices in collaborative applications. However, there are many restrictions so that a successful integration can be accomplished. The restrictions include the screen size of mobile devices, limited battery power and maintenance related to cache coherence. Nowadays, a great deal of collaborative applications considers only strongly coupled environments where disconnection is not a problem. The goal of this paper is to present a mechanism to deal with cache coherence which occur in weakly coupled environments, and the awareness in collaborative applications.*

**Resumo.** *Os avanços da computação móvel em adição as novas formas de conectividade permitem a integração de dispositivos móveis em trabalhos colaborativos. Entretanto, existem muitas restrições para que essa integração seja realizada com sucesso. Entre eles, podem-se citar o tamanho da tela dos dispositivos móveis, limitado poder de bateria e coerência de cache. Atualmente, a maior parte das aplicações colaborativas considera ambientes fortemente acoplados, onde a desconexão dos dispositivos móveis em geral não é um problema. O objetivo deste artigo é apresentar um mecanismo para tratar a coerência de cache que ocorre em ambientes fracamente acoplados e a percepção de colaboração em aplicações colaborativas.*

## 1. Introdução

A computação móvel tem emergido como uma nova disciplina no campo da Ciência da Computação. Como dispositivos móveis, podem-se citar telefones celulares, *Personal Digital Assistants* (PDAs) e computadores portáteis. Esses dispositivos são providos de uma capacidade de processamento, armazenamento, e comunicação cada vez maior. Dessa forma, um passo importante é proporcionar maior interação entre os usuários móveis.

A área de Trabalho Cooperativo Apoiado por Computador, em inglês *Computer-Supported Cooperative Work* (CSCW), indica uma variedade de tecnologias as quais permitem que equipes de trabalho cooperem eletronicamente. Um exemplo dessas tecnologias são os sistemas de *groupware* [Luff and Heath 1998]. Um *groupware* é um sistema baseado em computador que suporta dois ou mais usuários engajados em uma tarefa comum e que forneça uma interface para ambiente compartilhado [Ellis and Gibbs 1989].

O avanço das tecnologias relacionadas às redes sem fio e a proliferação de dispositivos móveis têm possibilitado o desenvolvimento de sistemas de *groupware* móvel.

Entretanto, a maior parte das ferramentas para trabalho cooperativo não atendem as restrições impostas pelos computadores portáteis e redes sem fio [Buszko et al. 2001]. Dentre essas restrições pode-se destacar: gerência de energia, fraca conectividade e interface limitada.

A ocorrência de desconexões é comum em ambientes sem fio, o que torna importante a redução da necessidade de comunicação durante o processo de cooperação. Nesse sentido, a replicação de dados é uma técnica chave para fornecer uma maior disponibilidade e desempenho em ambientes móveis [Saito and Shapiro 2005]. As réplicas, armazenadas localmente nos dispositivos móveis, são conhecidas na literatura como cache [Barbara and Imieliski 1994, Chung and Cho 1998, Saito and Shapiro 2005].

A replicação torna-se uma importante forma de reduzir a quantidade de informações transferidas, assim como a latência de acesso aos dados e o tráfego da rede. No entanto, a replicação apenas possibilita essas reduções se o cliente encontrar as informações de seu interesse armazenadas na cache de seu dispositivo móvel. Por outro lado, é importante ressaltar que, por causa das desconexões torna-se difícil verificar a validade da cache.

Métodos eficientes de coerência de cache são críticos para garantir que as aplicações em ambientes sem fio tenham um desempenho razoável [Yuen et al. 2000]. Entretanto, a maioria dos métodos atuais para coerência de cache em ambientes móveis não foram projetados para trabalhar com sistemas cooperativos [Barbara and Imieliski 1994, Chung and Cho 1998, Wang et al. 2004]. Adicionalmente, a maior parte dessas estratégias não contempla a percepção (*awareness*). A percepção é definida neste artigo como um sentimento de ilusão através do qual um usuário identifica a presença dos outros participantes e algumas de suas atividades no espaço de trabalho compartilhado.

Nesse contexto, o principal objetivo deste trabalho é propor uma técnica de manutenção da coerência de cache e provisão de percepção em um ambiente cooperativo computadorizado. Considera-se que esse ambiente suporte a mobilidade dos usuários através de uma interface de comunicação sem fio.

O artigo está organizado da seguinte forma. A seção 2 fornece uma breve visão sobre os trabalhos relacionados. A seção 3 contém a descrição da técnica proposta para prover a coerência de cache e percepção em aplicações de *groupware* móvel. A simulação da técnica e análise dos resultados obtidos está descrita na seção 4. Finalmente, na seção 5, apresentam-se as conclusões e trabalhos futuros.

## **2. Trabalhos Relacionados**

No contexto da computação móvel, pode-se definir cache como uma forma de replicar os dados mais relevantes na memória dos dispositivos móveis. Dessa forma, a utilização da cache permite que esses dispositivos realizem o processamento de seus dados de forma mais rápida e consumindo uma quantidade menor de energia.

Neste trabalho, assume-se que os dados a serem manipulados no *groupware* possam ser replicados nos dispositivos móveis. Com isso, são geradas múltiplas cópias de uma determinada informação. Esse comportamento introduz o problema da consistência entre essas cópias, também conhecido como coerência de cache. Dessa forma, os be-

nefícios da replicação dos dados nas unidades móveis (UMs) apenas podem ser obtidos se a coerência de cache for mantida.

Para manter a coerência de cache é necessário informar as UMs quando itens de dados de suas caches forem alterados. Uma maneira de informar as UMs sobre essas alterações é através da notificação de invalidação (NI) [Barbara and Imieliski 1994]. A NI é uma abordagem utilizada para que a cache seja reutilizada após desconexões. Nessa abordagem, um servidor dissemina notificações contendo as alterações/invalidações a serem realizadas na cache das UMs. Esse servidor, responsável pelas notificações, pode ser do tipo *stateful* ou *stateless*. O servidor *stateful* tem conhecimento de quais UMs estão conectadas a ele e dos dados armazenados localmente nelas. Dessa forma, ao ocorrer uma alteração, ele envia notificações de invalidação (NIs) diretamente para as UMs afetadas. O servidor *stateless* não tem conhecimento sobre as UMs conectadas a ele, assim como, não possui conhecimento do conteúdo dos dados dessas UMs. Nesse caso, as mensagens de invalidação são enviadas para todas as UMs.

A maior parte dos trabalhos relacionados à coerência de cache em ambientes móveis utiliza servidores *stateless* [Barbara and Imieliski 1994, Berkenbrock and Dantas 2005]. Isso acontece porque as abordagens de invalidações imediatas, como a *stateful*, não são praticáveis em ambientes onde os clientes se conectam e desconectam freqüentemente da rede sem fio. Nesse caso, o servidor tem dificuldade em manter-se informado sobre todos os clientes que possuem uma cópia de um determinado registro que tenha sido modificado. Wang [Wang et al. 2004] propõe uma técnica de coerência de cache híbrida, onde são herdadas as características positivas das abordagens *stateless* e *stateful*. Entretanto, essas técnicas não foram projetadas para sistemas de *groupware*. Geralmente, os trabalhos relacionados à coerência de cache em ambientes móveis preocupam-se com a redução da quantidade de mensagens trocadas e eficiência da coerência, no entanto, eles não levam em consideração a percepção e o desempenho do trabalho cooperativo.

No trabalho de Neyem [Neyem et al. 2006], é proposta uma técnica para permitir o compartilhamento e manipulação de dados em um ambiente de *Mobile Ad-hoc NETWORKS* (MANETs). A técnica considera manipulação *off-line* dos dados, fornecendo funcionalidades de sincronização através de um processo de reconciliação. Para facilitar a detecção de conflitos e suportar a reconciliação dos dados, utiliza-se um sistema de versão que gerencia as diferentes versões dos documentos. A manutenção da coerência é baseada em árvores de diferenciação e técnicas de *merging*. Contudo, o trabalho não apresenta informações de percepção.

Kirsch-Pinheiro [Kirsch-Pinheiro et al. 2005] apresenta um *framework* para fornecer percepção com base no contexto do usuário. O ambiente pode ser acessado a partir de dispositivos móveis onde, devido às restrições como tamanho de tela reduzido, existe a necessidade de filtragem das informações de percepção. Nesse trabalho, o contexto atual do usuário é utilizado para filtrar as informações que sejam pertinentes à unidade móvel (UM).

Outro trabalho relacionado à percepção é o de Papadopoulos [Papadopoulos 2006]. Ele investiga oportunidades para melhorar a percepção em ambientes móveis. Papadopoulos considera em seu estudo, ambientes com restrições de

processamento e desconexões transitórias da rede. Para isso, ele elaborou um CSCW síncrono e desenvolveu um protocolo que propõe um balanceamento entre consumo de energia e tempo de notificação. O objetivo do autor é propor mecanismos que forneçam percepção para colaboradores móveis de maneira eficiente em termos de tempo e consumo de energia. Tanto o trabalho de Papadopoulos quanto o de Kirsch-Pinheiro são interessantes no que diz respeito à percepção, entretanto, eles não fornecem suporte a coerência de cache.

### **3. Uma Técnica para Coerência de Cache e Percepção em Ambientes de Cooperação Móvel**

A forma de interação entre os participantes de um *groupware* pode ser síncrona ou assíncrona. Um *groupware* síncrono apresenta a idéia de simultaneidade no tempo, isto é, as notificações de eventos são imediatamente propagadas para todos os participantes. Em um *groupware* assíncrono os participantes interagem em momentos distintos, sendo necessário a existência de registros de mudanças de estados. Considerando um ambiente de cooperação móvel, com possibilidade de desconexão dos participantes, manter a sincronização dos dados geralmente é dispendioso e pode ser inapropriado. A abordagem proposta neste artigo utiliza um ambiente flexível. Os participantes conectados ao sistema trabalham de forma síncrona, enquanto os participantes desconectados podem continuar a cooperação de forma assíncrona. Dessa forma, o sistema possibilita diferentes estilos de trabalho cooperativo. Na seção 3.1 serão apresentadas as principais características do ambiente do *groupware* móvel proposto. A seção 3.2 apresenta a arquitetura do ambiente. A seção 3.3 descreve a técnica para tratar o problema da coerência de cache e percepção no ambiente de cooperação.

#### **3.1. Ambiente Experimental**

Em dispositivos móveis maiores, como *notebooks*, a mobilidade é limitada por fatores como peso e tamanho do equipamento. Para que o *groupware* suporte uma maior mobilidade dos usuários, no ambiente proposto, serão investigados problemas relacionados com pequenos dispositivos móveis, como PDAs.

No ambiente, cada dispositivo móvel possui capacidade de armazenar dados localmente. Conforme apresentado na seção 2, os dados das UMs podem ser atualizados através de servidores *stateful* ou de servidores *stateless*. Neste trabalho, optou-se pela utilização de um ambiente híbrido, mesclando características das abordagens *stateless* e *stateful*. Adicionalmente, o ambiente fornecerá suporte à percepção. Para fornecer dados de percepção, o ambiente permite que as UMs tenham conhecimento de informações relacionadas à conectividade e as atividades dos membros no grupo. A percepção das atividades realizadas segue o modelo de interface *What You See Is What I Did* (WYSIWID) [Marshall et al. 1991]. Nesse modelo, as interações entre os usuários ocorrem de forma assíncrona, pois a propagação das alterações para os demais usuários será adiada até que as condições, presentes no servidor do *groupware*, sejam estabelecidas.

O ambiente proposto representa as informações compartilhadas através de uma estrutura de dados do tipo grafo. Várias formas de representação que usam grafos podem ser adotadas, como por exemplo redes de Petri e diagramas de classe da *Unified Modeling Language* (UML). Adicionalmente, vários trabalhos utilizam grafos para de-

monstrar métodos de coerência em ambientes compartilhados [Correa and Marsic 2003, Neyem et al. 2006].

Apesar de o ambiente experimental considerar aplicações baseadas em representação de grafos, a técnica proposta neste artigo pode ser aplicada em outras aplicações existentes tais como controle de estoque e jogos. A técnica também pode ser considerada em classes de aplicações ainda não existentes ou aquelas que precisam satisfazer requisitos específicos. Uma classe de aplicação interessante é aquela que provê auxílio a reuniões presenciais: (i) para votação com requisito de anonicidade e confidencialidade, (ii) organizar grupos de interesse de maneira mais eficiente para realizar tarefas etc. A característica comum das aplicações é a percepção de ações, consistência de dados e mobilidade dos participantes.

O trabalho cooperativo é realizado a partir de um ambiente infra-estruturado, isto é, a rede sem fio envolve servidores com localização fixa (Estação de Suporte Móvel), que se comunicam com os dispositivos móveis (Unidades Móveis) via frequências de radio. A Estação de Suporte Móvel (ESM) possui um ponto de acesso IEEE 802.11 (interface sem fio). Através desse ponto de acesso, a ESM fica responsável pela área geográfica chamada célula. Apenas as UMs que estiverem na área de cobertura dessa célula poderão se comunicar com o servidor. Com isso, a ESM atua como um intermediário entre as UMs e a Estação Fixa (EF), sendo responsável pela comunicação. A EF é o servidor de *groupware*. A Figura 1 representa o ambiente experimental.

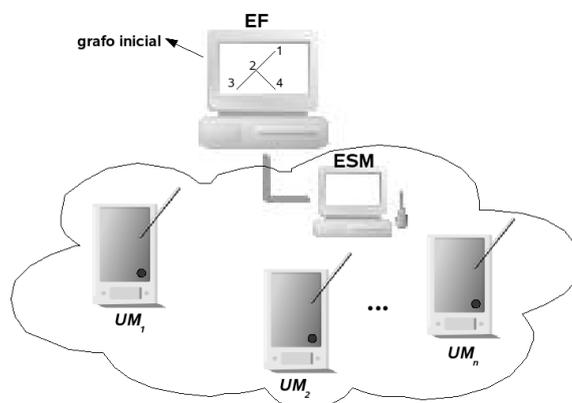


Figura 1. Ambiente experimental

### 3.2. Arquitetura do Ambiente

A arquitetura do ambiente proposto é apresentada na Figura 2. Os componentes principais dessa arquitetura são a EF e as UMs. Nesse ambiente, a EF se comunica com as UMs através do canal de *downlink*. Essa comunicação apenas será realizada se a UM possuir energia e estiver na área de cobertura da rede sem fio. As UMs podem obter réplicas dos grafos armazenados na EF. Adicionalmente, elas podem solicitar alteração dos grafos através do canal de *uplink*. No entanto, essa alteração apenas será realizada com a confirmação da EF.

A EF é composta pelos módulos descritos a seguir:

- Gerenciador de *groupware* (GG) - Responsável pelo gerenciamento das UMs no *groupware*. Através do GG, a EF recebe solicitações das UMs. O GG também é

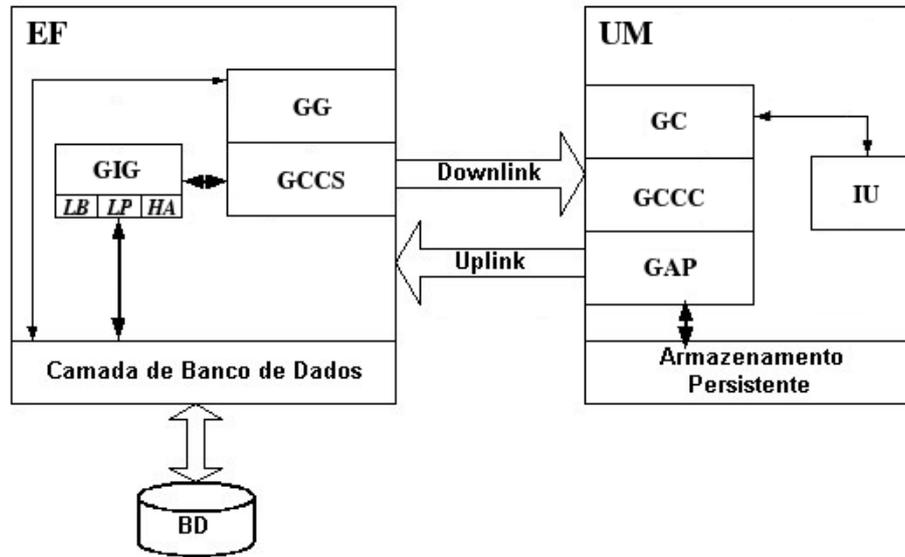


Figura 2. Arquitetura do ambiente

responsável por retornar as respostas a essas solicitações e enviar as notificações dos eventos ocorridos;

- Gerenciador para coerência de cache no servidor (GCCS) - Realiza o gerenciamento da coerência de cache e da percepção na EF. É responsável pela atualização do rótulo de tempo da EF ( $RT_{EF}$ ), rótulo de tempo das UMs ( $RT_{UM}$ ), janela de desconexão ( $w$ ) e contador de tempo de desconexão ( $ct$ ). Essas variáveis possuem papel importante na realização da coerência de cache e provisão da percepção. Os valores  $RT_{UM}$  e  $RT_{EF}$  ajudam a verificar se as UMs deixaram de receber alguma notificação de alteração ( $NA$ ). A  $NA$  contém as informações necessárias para que as UMs atualizem suas caches, sempre que ocorrer alguma alteração no grafo. Assim, a  $NA$  auxilia no fornecimento da percepção a respeito das atividades realizadas pelos participantes do *groupware*. Os rótulos de tempo indicam quando o grafo foi atualizado pela última vez. A atualização dos valores  $RT_{UM}$  e  $RT_{EF}$  ocorre a cada alteração do grafo na EF. O valor de  $w$  indica quanto tempo as UMs podem permanecer desconectadas, sem que seus dados sejam invalidados. Os valores de  $w$  e  $ct$  são utilizados no fornecimento de um *time-out* para liberar bloqueios (travas) em itens de dados solicitadas por UMs desconectadas. Adicionalmente, o GCCS é responsável por ativar o gerenciador de informações no *groupware* (GIG) quando for necessário realizar alguma alteração no banco de dados; e
- Gerenciador de informações no *groupware* (GIG) - Responsável pela alteração da lista de bloqueios ( $LB$ ), lista de pendências ( $LP$ ) e histórico de alterações ( $HA$ ). As informações de bloqueios de nodos do grafo, realizados pelo grupo, são armazenados na  $LB$ . A  $LB$  será utilizada para verificar se as caches das UMs, que se desconectaram da rede, precisarão ser invalidadas. A  $LP$  contém informações sobre as UMs que desejam trabalhar em um nodo que já foi bloqueado. O  $HA$  contém as alterações realizadas nos últimos  $w$  segundos. Essas variáveis são fundamentais tanto para o fornecimento da coerência de cache e percepção quanto para possibilitar a interação assíncrona. Por exemplo, com as informações conti-

das no *HA* e na *LP*, as UMs terão conhecimento das atividades realizadas pelos outros participantes do *groupware*. Adicionalmente, se uma UM ficar desconectada, ela poderá atualizar sua cache com o auxílio das informações do *HA* e da *LB*.

Cada UM é composta pelos seguintes módulos:

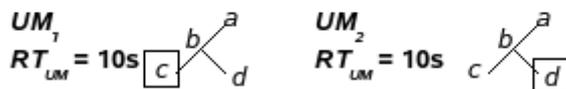
- Gerenciador de conexão (GC) - Gerencia a conexão da UM com a EF;
- Gerenciador para coerência de cache no cliente (GCCC) - Nele ficam registradas o valor do rótulo de tempo da sua UM ( $RT_{UM}$ ), janela de desconexão ( $w$ ) e contador de tempo de desconexão ( $ct$ ). Os valores de  $RT_{UM}$  e  $w$  são atualizados pela EF. O valor de  $ct$  indica à quanto tempo a UM está desconectada. O GCCC também é responsável por ativar o gerenciador do armazenamento persistente (GAP) quando for necessário realizar alguma alteração na cache da UM;
- Gerenciador do armazenamento persistente (GAP) - Executa as alterações necessárias na cache do dispositivo móvel; e
- Interface com o usuário (IU) - Representa uma interface amigável para que as UMs possam acessar o ambiente. Através dessa interface a UM se mantém informada sobre a atividade dos outros participantes do *groupware* e pode realizar consultas/alterações em suas réplicas locais, e confirmá-las na EF.

### 3.3. Descrição da Técnica

Os seguintes passos constituem uma descrição informal da técnica para coerência de cache e percepção em sistemas de *groupware* com suporte a mobilidade:

1. Após entrarem no *groupware*, as UMs solicitam o envio do grafo relacionado ao seu grupo cooperativo. Dessa forma, o grafo é armazenado na cache de cada UM. Adicionalmente, a EF envia um rótulo de tempo que será armazenado nas UMs para consistências futuras. Então, as UMs podem solicitar o bloqueio (trava) dos nodos de interesse;
2. Ao receber uma solicitação de bloqueio da UM, a EF pode aceitá-la ou recusá-la. Em caso de aceite, a EF notifica os demais membros do grupo sobre o bloqueio. Os bloqueios realizados pelo grupo são armazenados na *LB*. Se a EF recusar o bloqueio, ela envia uma mensagem para a UM solicitante com o motivo da recusa. Se a recusa ocorre porque o nodo solicitado já está bloqueado, a EF envia uma notificação para a UM que possui o bloqueio, avisando sobre o interesse do outro participante do grupo. A EF, então, adiciona a UM solicitante na *LP*. A UM apenas começa a trabalhar no grafo, após conseguir o bloqueio de algum nodo. Utiliza-se a abordagem centralizada de prevenção de *deadlock* para evitar o *deadlock* de recurso. As UMs devem solicitar o bloqueio de todos os nodos desejados de uma única vez. A EF tem o conhecimento do estado global da alocação dos nodos através da *LB*;
3. Cada UM portadora de um grafo possui o  $RT_{UM}$ , enviado pela EF, indicando quando o grafo foi atualizado pela última vez, conforme o exemplo apresentado na Figura 3. Após alterar o grafo, a UM pode confirmar as alterações com a EF. Antes de efetivar as alterações, a EF faz as seguintes verificações:
  - Se  $RT_{UM} = RT_{EF}$ , a EF registra as alterações da UM no *HA*. Em seguida, a EF atualiza  $RT_{EF}$  (o novo valor indicará o instante dessa última

$UM_1$  bloqueia nodo  $c$   
 $UM_2$  bloqueia nodo  $d$   
 EF possui valor de  $w = 50s$  e  $RT_{EF} = 10s$



**Figura 3. UMs solicitam bloqueio de parte do grafo**

$UM_1$  desconecta da rede  
 $UM_2$  atualiza dados na EF, o novo valor de  $RT_{EF} = 40s$   
 EF atualiza HA e desbloqueia nodo  $d$



**Figura 4.  $UM_1$  desconecta da rede e  $UM_2$  confirma alteração de dados**

alteração) e envia uma *NA* para demais UMs, pertencentes ao grupo, que estão conectadas. A *NA* contém os novos dados do grafo e o novo rótulo de tempo (para que as UMs atualizem o valor  $RT_{UM}$  com o novo valor do  $RT_{EF}$ ). Nesse momento, o bloqueio do nodo que foi atualizado é liberado. Veja o exemplo na Figura 4. A EF, então, verifica na *LP* se existe alguma UM interessada em obter bloqueio do nodo liberado. Caso positivo, a EF envia uma notificação para a UM interessada, questionando sobre o interesse em alterar o nodo liberado;

- Se  $RT_{EF} - RT_{UM} \leq w$ , a EF envia para a UM uma mensagem contendo todas as alterações realizadas após o tempo  $RT_{UM}$ . Com isso, a UM pode atualizar os seus dados e reiniciar o processo de confirmação da alteração;
  - Se  $RT_{EF} - RT_{UM} > w$ , isso significa que ocorreram alterações no *groupware* que já não estão mais armazenadas no histórico. A EF, então verifica na *LB* se algum outro participante bloqueou o nodo utilizado pela UM após o tempo  $RT_{UM}$ . Caso positivo, a EF não tem capacidade de verificar a coerência de cache. Sendo assim, será necessário invalidar as informações da UM. Caso contrário, a EF envia novamente o grafo para que a UM possa unir as suas alterações com o grafo atual;
4. Quando uma UM se desconecta da célula de comunicação, se ela possuir algum nodo bloqueado, a EF avisa as demais UMs sobre a desconexão. Nesse caso, a partir do momento de desconexão, a EF e a UM desconectada iniciam o contador *ct*. É necessário que *ct* exista tanto na UM quanto na EF, pois é importante que a UM, ainda que desconectada, tenha percepção de quanto tempo ela poderá trabalhar. Adicionalmente, a EF precisa ter conhecimento do tempo de desconexão da UM. Enquanto *ct* for menor do que  $w$ , a EF não permite o bloqueio a nenhuma outra UM do nodo que está sendo atualizado pela UM desconectada. Caso contrário, a EF liberará o nodo bloqueado;
  5. Quando a UM se reconecta a célula de comunicação, é verificado o valor de *ct*:
    - Se  $ct \leq w$ , a UM desconectada pode solicitar confirmação de alteração com a EF. Porém, antes disso, a EF verificará o  $RT_{UM}$  e enviará uma mensagem contendo todas as alterações que ocorreram no grupo enquanto a UM permaneceu desconectada. Veja o exemplo na Figura 5. Com isso, a UM atualiza os seus dados e reinicia o processo de confirmação da alteração, conforme o exemplo mostrado na Figura 6;
    - Se  $ct > w$  será necessário verificar, na *LB*, se outro participante bloqueou o nodo utilizado pela UM, durante o seu período de desconexão. Caso

existam registros de bloqueio do nodo na *LB*, será necessário invalidar as informações da UM. Caso contrário, a EF envia novamente o grafo para que a UM realize a união das suas alterações ao grafo atual.

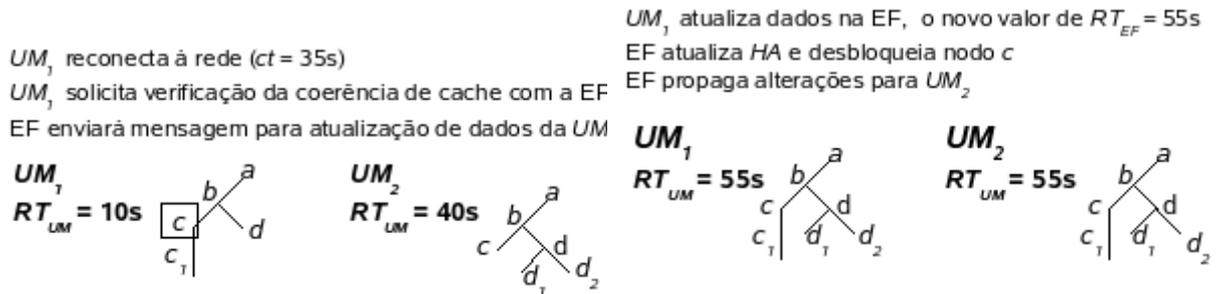


Figura 5.  $UM_1$  reconecta à rede

Figura 6.  $UM_1$  confirma alteração de dados

#### 4. Avaliação da Técnica

Durante o processo de cooperação, as UMs consomem energia para enviarem e receberem dados. Dessa forma, para que uma UM possa permanecer uma quantidade maior de tempo conectada, é importante reduzir a quantidade de mensagens trocadas durante o processo de cooperação [Papadopoulos 2006]. Os experimentos conduzidos neste artigo, consideram a quantidade de dados enviados e recebidos para verificar a eficiência da estratégia proposta.

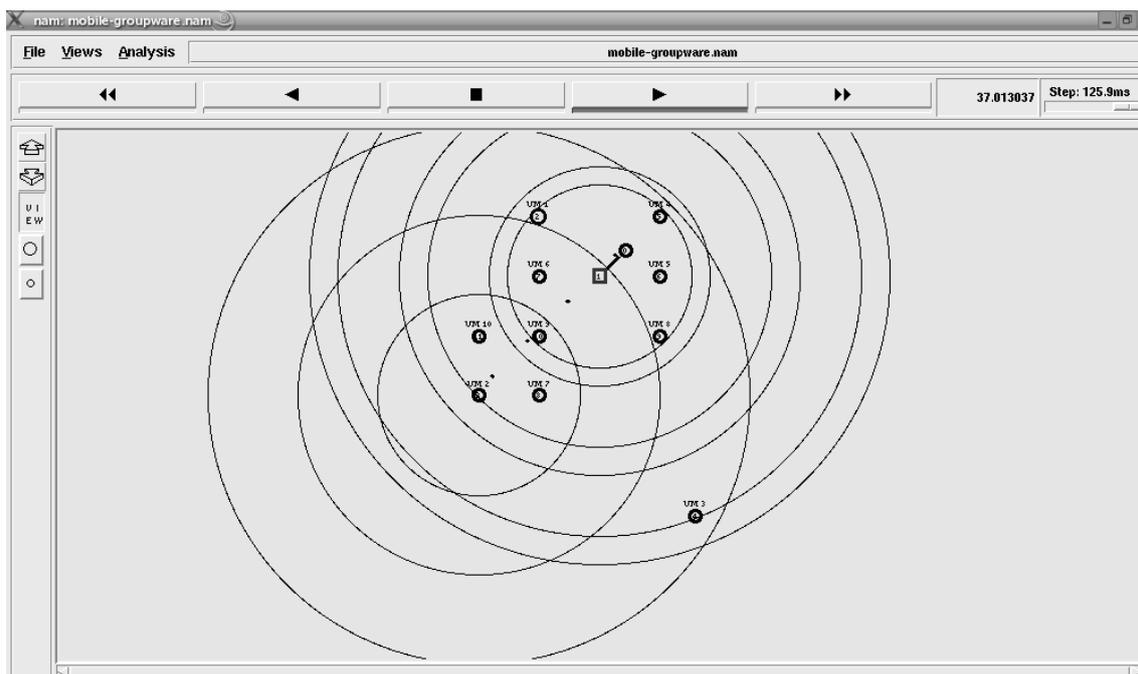
Os experimentos foram conduzidos utilizando o simulador *Network Simulator* (NS-2) [Berkeley et al. 2006]. O NS-2 é um simulador para eventos discretos direcionado à pesquisa em redes de computadores. O NS-2 possibilita a visualização gráfica dos dados simulados através da ferramenta chamada *Network Animator* (NAM). A Figura 7 mostra animação do ambiente projetado. O nodo na forma de quadrado representa a ESM. O nodo conectado diretamente a ele corresponde a EF. Os demais nodos são as UMs.

O modelo desenvolvido suporta tanto redes cabeadas (para comunicação da EF com a ESM) quanto redes sem fio (para comunicação das UMs com a ESM). Ele foi escrito na linguagem de programação *Tool Command Language* (Tcl).

Para simular a técnica proposta para coerência de cache e percepção foi desenvolvida uma aplicação chamada *Mobile Groupware Cache Coherence* (MGCC). Essa aplicação foi escrita utilizando a linguagem de programação C++. O modelo, implementado em Tcl, em conjunto com a aplicação MGCC foram definidos de acordo com a arquitetura apresentada na seção 3.2. A aplicação MGCC é inserida em cada UM.

No ambiente é utilizado o protocolo de rede UDP e a camada MAC de acordo com o padrão 802.11. O link entre a EF e a ESM é bi-direcional e possui capacidade de 5 Mb/s para cada direção. O MGCC foi utilizado em um cenário com área de simulação de 400m×400m, contendo 10 nodos móveis. Nesse cenário ocorre a média de uma alteração por minuto.

O valor de  $w$ , considerado nos experimentos, é de 65 segundos. A configuração da máquina utilizada para simular o ambiente de cooperação é: processador AMD Sempron 2300, 512 MB de memória e sistema operacional Open Suse Linux 10.1.

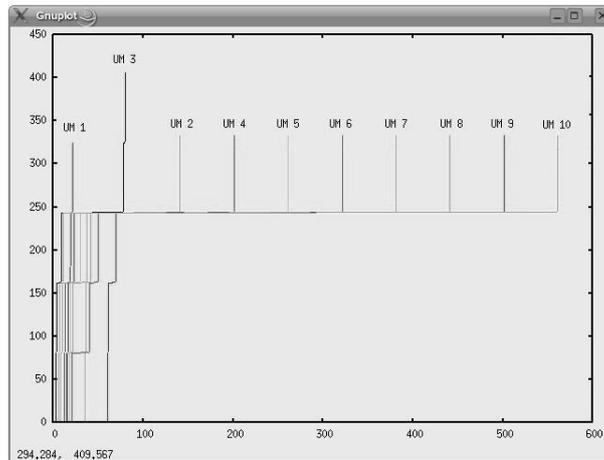


**Figura 7. Animação do experimento**

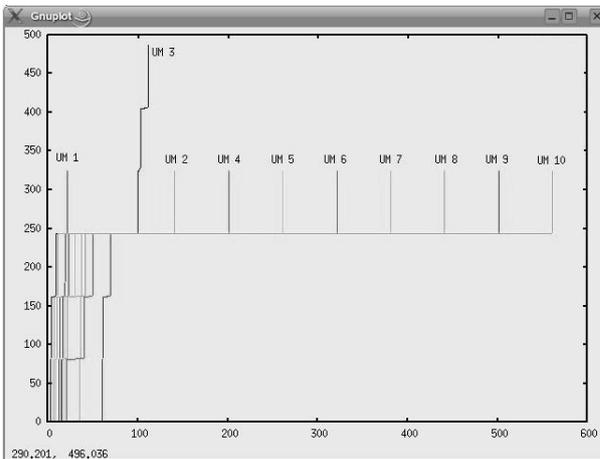
As Figuras 8, 9 e 10 apresentam os dados trocados entre as UMs e a EF, em um experimento contendo apenas uma desconexão. Nas Figuras 11, 12 e 13 são mostradas as mensagens trocadas no sistema, em um experimento com 3 desconexões. As desconexões apresentadas nas Figuras 8 e 11 possuem tempo inferior a  $w$ . Nesse caso, para garantir a coerência de cache e percepção, foi necessária a troca de um maior número de mensagens entre as UMs que ficaram desconectadas e a EF (média de 405 pacotes), se comparado com as UMs que permaneceram conectadas durante todo o experimento (média de 324 pacotes).

Quando o tempo de desconexão é maior do que  $w$ , poderão ocorrer duas situações. Na primeira situação, nenhum outro participante bloqueou o nodo utilizado pela UM durante a sua desconexão. Nesse caso, a EF envia novamente o grafo para a UM unir suas alterações ao grafo atual (Figuras 9 e 12). Na segunda situação, existe alguma outra UM na *LB*, bloqueando o nodo que a UM desconectada estava atualizando. Nesse caso, os dados da UM que acabou de reconectar serão invalidados (Figuras 10 e 13). Se a UM que teve os seus dados invalidados desejar continuar participando das atividades do grupo, ela necessitará reiniciar o processo de entrada no grupo cooperativo, o que resultará em um aumento no número de mensagens trocadas no sistema MGCC. Nessas duas situações, a média de pacotes trocados entre as UMs desconectadas e a EF é 486. Contudo, na segunda situação ocorre um aumento no número de mensagens trocadas não apenas entre as UMs que ficaram desconectadas, como também nas UMs que adquiriram os novos bloqueios.

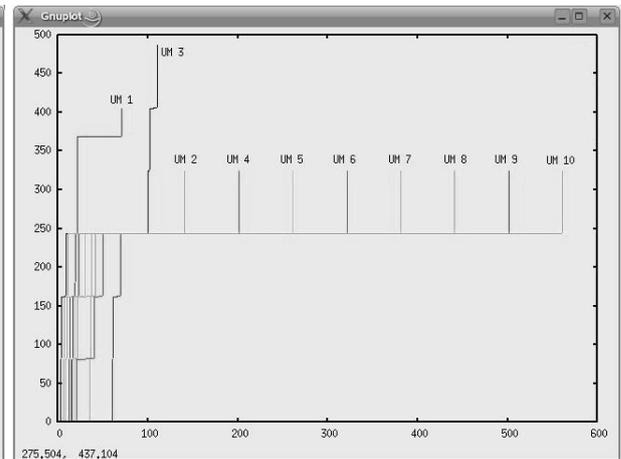
A técnica proposta possui melhor desempenho em ambientes com um número reduzido de desconexões. Adicionalmente, o ambiente mostra-se eficiente em situações onde os intervalos de desconexão são menores do que  $w$ . Quando os intervalos de desconexão são maiores do que  $w$ , se outras UMs adquirem os bloqueios das UMs desconectadas, ocorrem invalidações dos dados armazenados nas UMs. Essas invalidações



**Figura 8. Desconexão de uma UM com  $ct$  inferior a  $w$**



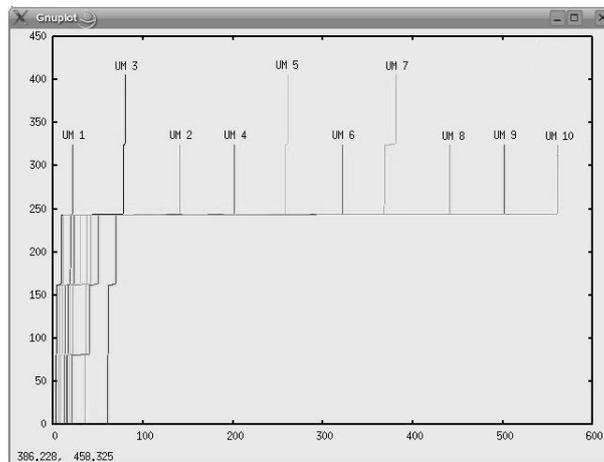
**Figura 9. Desconexão de uma UM com  $ct$  superior a  $w$ , sem novos bloqueios no nodo em alteração**



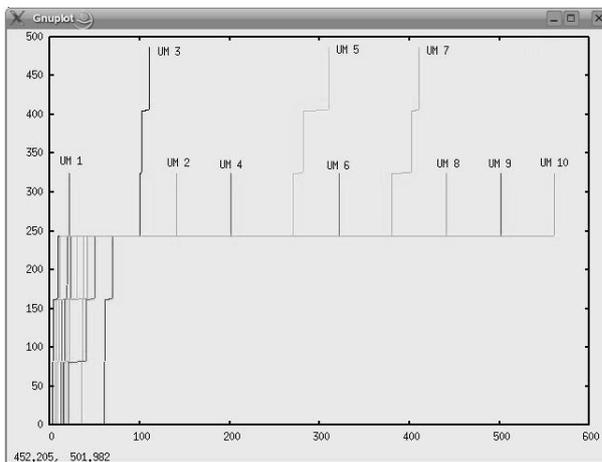
**Figura 10. Desconexão de uma UM com  $ct$  superior a  $w$ , com novo bloqueio no nodo em alteração**

são necessárias para assegurar a coerência de cache. Entretanto, um número elevado de invalidações comprometerá o desempenho do sistema, ocasionando retrabalhos e desmotivando os participantes do *groupware*. Em contrapartida, para evitar essas invalidações, a técnica usa a *LB*. Com a *LB*, o número de invalidações é reduzido, entretanto, a quantidade de mensagens trocadas entre a EF e a UM continua alta. Isso acontece porque, nesse caso, a EF já não possui na *HA* todas as alterações realizadas no grupo cooperativo durante o período de desconexão da UM. Então, para manter a cache, a EF necessita propagar todo o conteúdo do grafo para que a UM possa incluir no grafo as alterações que foram efetuadas em sua cache local.

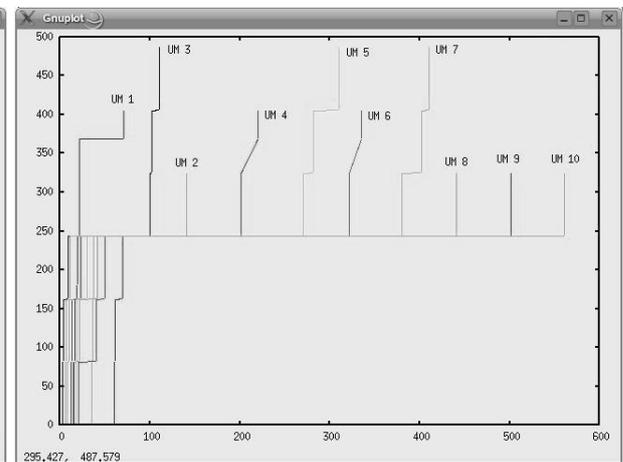
Analisando a técnica proposta, pode-se provar informalmente que a coerência de cache é garantida e livre de *deadlock*. Isso decorre pelas seguintes razões: (i) uso de bloqueios para realizar o controle de concorrência, (ii) uso de *time-out* para liberar bloqueios em itens de dados solicitados por UMs desconectadas, (iii) carga do grafo atualizado para as UMs desconectadas acima do período de *time-out*, e (iv) uso da abordagem centralizada



**Figura 11. Desconexão de três UMs com  $ct$  inferior a  $w$**



**Figura 12. Desconexão de três UMs com  $ct$  superior a  $w$ , sem novos bloqueios nos nodos em alteração**



**Figura 13. Desconexão de três UMs com  $ct$  superior a  $w$ , com novos bloqueios nos nodos em alteração**

de prevenção de *deadlock*.

É importante ressaltar que a definição de um valor de  $w$  ideal, depende do tipo de aplicação. Ambientes com um valor de  $w$  pequeno podem resultar em elevado número de invalidações. No entanto, um valor alto para  $w$  pode resultar em retardos para obtenção de bloqueios e aumento da troca de mensagens no sistema. Dessa maneira, acredita-se que seja essencial obter um conhecimento prévio sobre comportamento dos usuários móveis dentro *groupware*, para que se possa determinar um valor eficiente para a janela de desconexão.

## 5. Conclusões e Trabalhos Futuros

Uma infra-estrutura completa para o desenvolvimento de sistemas de *groupware* móvel envolve uma série de fatores, como interface, acoplamento e interação. Cada um desses aspectos constitui áreas de pesquisas por si só. Neste artigo foi apresentado o projeto e simulação de uma estratégia para tratar os problemas de manutenção da coerência de

cache e falta de percepção em ambientes de *groupware* móveis.

Os experimentos foram conduzidos utilizando o simulador *Network Simulator*. Nele foi desenvolvido um modelo que representa a estrutura de um cenário de uso proposto. Adicionalmente, foi desenvolvida uma aplicação chamada *Mobile Groupware Cache Coherence* (MGCC).

Trabalhos futuros incluem o aperfeiçoamento do MGCC possibilitando a simulação de cenários de uso mais complexos. Pretende-se possibilitar que o sistema identifique o tamanho da janela de desconexão ( $w$ ) de forma adaptativa. Assim, com base em informações relacionadas às desconexões e quantidade de bloqueios, o sistema poderá alterar o valor de  $w$ , reduzindo o tráfego na rede e a quantidade de invalidações.

Adicionalmente, pretende-se implementar o MGCC em um ambiente real. Para isso, será importante definir técnicas de visualização e filtragem de informações de percepção, considerando que o tamanho da tela dos PDAs dificulta a manipulação das informações.

## 6. Agradecimentos

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq pelo apoio financeiro. Número do processo: 141781/2006-8. Agradecemos também aos revisores anônimos pela importante colaboração através dos comentários e sugestões.

## Referências

- Barbara, D. and Imieliski, T. (1994). Sleepers and workaholics: caching strategies in mobile environments. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, NY, USA. ACM Press.
- Berkeley, U., LBL, USC/ISI, and PARC., X. (2006). The ns manual.
- Berkenbrock, C. D. M. and Dantas, M. A. R. (2005). Investigation of cache coherence strategies in a mobile client/server environment. *International Conference on Computational Science (ICCS 2005)*.
- Buszko, D., Lee, W.-H. D., and Helal, A. S. (2001). Decentralized ad-hoc groupware api and framework for mobile collaboration. In *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pages 5–14, New York, NY, USA. ACM Press.
- Chung, H. and Cho, H. (1998). Data caching with incremental update propagation in mobile computing environments. *Australian Computer Journal*, 30(2):77–86.
- Correa, C. D. and Marsic, I. (2003). Software framework for managing heterogeneity in mobile collaborative systems. pages 125–134.
- Ellis, C. A. and Gibbs, S. J. (1989). Concurrency control in groupware systems. In *SIGMOD '89: Proceedings of the 1989 ACM SIGMOD international conference on Management of data*, pages 399–407, New York, NY, USA. ACM Press.
- Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., and Martin, H. (2005). Bw-m: a framework for awareness support in web-based groupware systems. In *Proceedings of*

- the Ninth International Conference Computer Supported Cooperative Work in Design, 2005*, pages 240–246. IEEE Computer Society.
- Luff, P. and Heath, C. (1998). Mobility in collaboration. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 305–314, New York, NY, USA. ACM Press.
- Marshall, C. C., Halasz, F. G., Rogers, R. A., and William C. Janssen, J. (1991). Aquanet: a hypertext tool to hold your knowledge in place. pages 261–275.
- Neyem, A., Ochoa, S. F., and Pino, J. A. (2006). A strategy to share documents in manets using mobile devices. In *The 8th International Conference Advanced Communication Technology, ICACT 2006*, pages 1400–1404. IEEE Computer Society.
- Papadopoulos, M.-C. (2006). Improving awareness in mobile cscw. *IEEE Transactions on Mobile Computing*, 5(10):1331–1346. Member-Constantinos Papadopoulos.
- Saito, Y. and Shapiro, M. (2005). Optimistic replication. *ACM Comput. Surv.*, 37(1):42–81.
- Wang, Z., Das, S. K., Che, H., and Kumar, M. (2004). A scalable asynchronous cache consistency scheme (saccs) for mobile environments. In *IEEE transactions on parallel and distributed systems*. IEEE Computer Society.
- Yuen, J. C.-H., Chan, E., Lam, K.-Y., and Leung, H. W. (2000). Cache invalidation scheme for mobile computing systems with real-time data. *SIGMOD Rec.*, 29(4):34–39.

# Utilização de RPG e MABS no Desenvolvimento de Sistemas de Apoio à Decisão em Grupos

Diana F. Adamatti<sup>1</sup>, Jaime S. Sichman<sup>1</sup>, Helder Coelho<sup>2</sup>

<sup>1</sup>Laboratório de Técnicas Inteligentes  
Escola Politécnica - Universidade de São Paulo - Brasil

<sup>2</sup>Laboratório de Modelação de Agentes  
Faculdade de Ciências - Universidade de Lisboa - Portugal

{diana.adamatti, jaime.sichman}@poli.usp.br, hcoelho@di.fc.ul.pt

**Abstract.** *This paper proposes the integrated use of RPG (Role-Playing Games) and MABS (Multi-Agent-Based Simulation) techniques to develop Group Decision Support Systems (GDSS), using GMABS methodology. Natural resources management was the chosen domain to implement GDSS, because in this domain negotiation process is extremely important and complex. Two prototypes are presented, JogoMan and ViP-JogoMan, showing that GMABS methodology can be applied to GDSS and can bring excellent results in negotiation process.*

**Resumo.** *Este artigo propõe a utilização integrada das técnicas de RPG (Jogos de Papéis) e MABS (Simulação Baseada em Multiagentes) para desenvolvimento de Sistemas de Apoio à Decisão em Grupos (SAD-G), a partir da metodologia GMABS. O domínio escolhido para desenvolvimento do SAD-G foi o de gestão de recursos naturais, visto que neste domínio o processo de negociação é extremamente importante e complexo. Dois estudos de caso são apresentados, os protótipos JogoMan e ViP-JogoMan, mostrando que a metodologia GMABS pode ser aplicada para SAD-G e trazer excelentes resultados no processo de negociação.*

## 1. Introdução

Jogos de Papéis (Role-Playing Games (RPGs)) são um tipo de jogo em que os jogadores "interpretam" uma personagem, criada dentro de um determinado cenário (ambiente). As personagens respeitam um sistema de regras, que serve para organizar suas ações, determinando os limites do que pode ou não ser feito [Klimick 2003]. RPG é uma técnica muito utilizada em treinamento, pois pode colocar os jogadores em situações de tomada de decisão sem conseqüências reais. Em especial, grandes empresas têm utilizado RPG em cursos de treinamento devido ao fator lúdico envolvido nos jogos, o que faz com que o treinamento e/ou aprendizagem de determinado assunto seja facilitado.

Já os chamados Sistemas Multiagentes estudam o comportamento de um conjunto de agentes autônomos, eventualmente com características diferentes, evoluindo em um ambiente comum. Estes agentes podem interagir uns com os outros, com o objetivo de realizar suas tarefas de modo cooperativo, compartilhando informações, evitando conflitos e coordenando a execução de atividades [Alvares and Sichman 1997]. Adicionalmente, a utilização da simulações como elemento auxiliar na tomada de decisões humanas tende a ser muito eficaz, pois seu emprego permite o exame de detalhes específicos com grande precisão [Strack 1984]. Da integração das tecnologias de

agentes e de simulação, surgiu a área de Simulação Baseada em Multiagentes (Multi-Agent-Based Simulation (MABS)), que é especialmente útil em domínios em que a interdisciplinaridade se faz presente [Gilbert and Troitzsch 1999]. Um destes domínios é a gestão de recursos naturais, área pública de extrema importância, que atualmente vem chamando muita atenção de diversos órgãos de pesquisa e de instituições públicas e privadas. Tal gestão pode ser considerada um problema complexo, visto que abrange diversas áreas de conhecimento e pesquisa, como sociologia, hidrologia e biologia. A utilização de MABS e RPG (de forma isolada ou até mesmo combinada) tem sido explorada em diversos trabalhos relacionados ao gerenciamento de recursos naturais [Barreteau et al. 2004, Dare and Barreteau 2004, D’Aquino et al. 2003], e pode trazer resultados interessantes, pois une a capacidade dinâmica de MABS com a capacidade de discussão e aprendizado das técnicas de RPG. A utilização integrada das técnicas de RPG e MABS será denominada neste trabalho de **Games and Multi-Agent-Based Simulation (GMABS)**.

Os Sistemas de Apoio a Decisão (SADs) são sistemas de informações, que além de fornecer informações sobre o domínio do problema em estudo, devem contribuir para o processo de tomada de decisão [Dornelas 2000]. Um sub-grupo dos SADs são os Sistemas de Apoio a Decisão em Grupos (SAD-Gs), onde o enfoque está trabalhar a tomada de decisão para grupos de pessoas. No trabalho cooperativo, normalmente, existem problemas de coordenação e tempo consumido para executar as tarefas. Sistemas implementados a partir da metodologia GMABS são uma classe de SAD-G.

Desta maneira, o objetivo deste artigo é apresentar a implementação de um SAD-G utilizando as técnicas de RPG e MABS, baseado na metodologia GMABS para o domínio da gestão de recursos naturais. Nas seções 2 e 3 são apresentadas as duas técnicas utilizadas neste trabalho, RPG e MABS, respectivamente. A seção 4 apresenta a metodologia definida para utilização de forma integrada das duas técnicas anteriormente citadas, denominada GMABS. Na seção 5 são apresentados os protótipos desenvolvidos a partir desta metodologia, para o domínio da gestão de recursos naturais. Finalmente, as conclusões deste trabalho e os trabalhos futuros propostos encontram-se na seção 6.

## **2. Jogos de Papéis (RPG)**

Os Jogos de Papéis (Role-Playing Games (RPGs)) estão situados entre os jogos e o teatro e consistem numa técnica onde se determinam regras e comportamentos de jogadores, bem como um contexto imaginário (ambiente). Desta maneira, RPGs são jogos onde cada jogador desempenha um papel e toma decisões, a fim de alcançar seus objetivos. Na verdade, os jogadores utilizam RPG como um "laboratório social", isto é, como uma forma de experimentar uma variedade de possibilidades, sem sofrer as conseqüências do mundo real [Barreteau et al. 2003].

RPGs podem ser impressos (mapas, fichas de personagens, etc.), eletrônicos ou orais. Consistem em uma categoria à parte nos jogos, em função de seu aspecto de colaboração em vez de competição: os RPGs não são jogos em que se tem um final com ganho ou perda. Ao final, deve-se completar uma história construída a partir das regras do jogo, buscando objetivos individuais e/ou coletivos. Desta forma, um fator muito marcante dos RPGs é sua capacidade de integração, visto que é jogado em grupos, e como dito acima é voltado para a cooperação entre seus participantes. Além disso, é um jogo em que o discurso, diálogo e troca de idéias são vitais para o seu desenvolvimento. Existe

um famoso ditado no meio do RPG que sintetiza esta filosofia: "grupos separados levam a mortes simultâneas" [Klimick 2003]. Desta maneira, os RPGs têm como foco a interação entre os indivíduos, ou seja, seus jogadores.

RPGs possuem o potencial de, através do exercício da fantasia, agir positivamente no desenvolvimento mental do homem e, conseqüentemente, no seu desenvolvimento social. Se observado com maior cuidado, pode-se perceber a força de integração latente de auxílio pedagógico, pois o jogo estimula uma troca constante de informações e experiências. Assim, "se bem direcionado e explorado, o RPG tem tudo para ter um papel marcante na sociedade"[Klimick 2003].

### **3. Simulação Baseada em Multiagentes (MABS)**

Da união das tecnologias de Simulação e Sistemas Multiagentes, surge a área de Simulação Baseada em Multiagentes (Multi-Agent-Based Simulation(MABS)). Muitas aplicações em sistemas multiagentes são desenvolvidas para simular alguma situação da realidade. Em MABS, o fenômeno real é decomposto em um conjunto de elementos e em suas interações. Cada elemento é modelado como um agente e o modelo geral é o resultado das interações entre estes agentes.

Modelar um fenômeno sob a perspectiva de um sistema multiagentes pode ser visualizado de acordo com as seguintes etapas [Frozza 1997]:

- decompor o fenômeno em um conjunto de elementos autônomos;
- modelar cada um dos elementos como um agente, definindo seu conhecimento, funções, comportamento e modos de interação;
- definir o ambiente dos agentes;
- definir quais agentes possuem a capacidade de ação e comunicação.

MABS é especialmente útil na união de diferentes perspectivas interdisciplinares, provenientes de várias áreas científicas, como psicologia social, ciência da computação, biologia social, sociologia e economia [Gilbert and Troitzsch 1999]. O caráter interdisciplinar de MABS é um aspecto muito importante, pois permite analisar diferentes teorias, metodologias, terminologias e pontos de vista [Marietto et al. 2002].

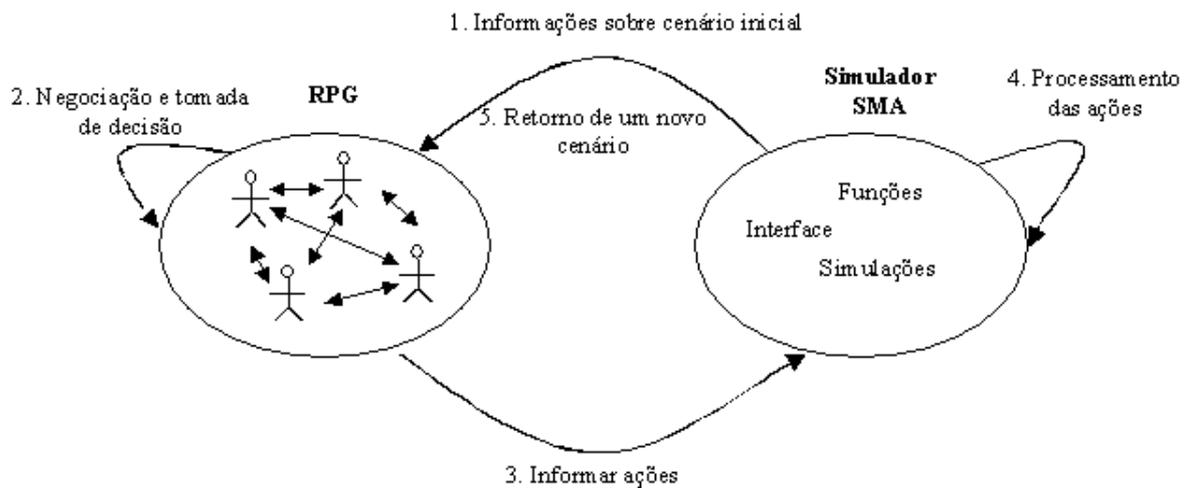
Segundo Drogoul e Ferber [Drogoul and Ferber 1992], os objetivos de MABS são:

- *testar* hipóteses sobre a emergência de comportamentos no nível macro a partir de interações no nível micro;
- *construir* teorias que contribuam para o entendimento de fenômenos etológicos, sociológicos ou psico-sociais, que relacionam comportamentos a estruturas;
- *integrar* teorias parciais de diferentes disciplinas (sociologia, psicologia cognitiva, etologia) numa mesma estrutura.

Para Doran [Doran 2001], a simulação baseada em multiagentes é uma grande promessa para habilitar a sociedade ao trabalho com modelos computacionais, de forma a avaliar os impactos cognitivos, tanto no âmbito individual quanto no coletivo, de sistemas sociais.

### **4. Metodologia GMABS**

A metodologia de utilização integrada das técnicas de RPG e MABS foi proposta inicialmente em [Barreteau et al. 2001] e será aqui denominada *Games and Multi-Agent-Based Simulation* (GMABS). O funcionamento desta metodologia segue a seqüência de passos apresentada na Figura 1:



**Figura 1. Funcionamento da Metodologia GMABS.**

1. Os jogadores recebem todas as informações sobre o jogo (regras e cenário inicial). São definidos os papéis que cada jogador irá desempenhar. Suponha um jogo em que existam os papéis de industrial e ecologista, e cujo objetivo é verificar a qualidade das águas de uma determinada região. Cada jogador primeiramente saberá quais as regras que cada papel pode executar e os possíveis benefícios e/ou prejuízos que estas ações podem causar a qualidade da água, bem como onde estão fisicamente localizados no jogo e que recursos possuem inicialmente (dinheiro, terras, etc.). Para o caso do papel do industrial, pode ser definido qual é o tamanho, local, lucratividade, quantidade de poluição, etc., de sua(s) empresa(s);
2. De posse de todas as informações necessárias para iniciar as negociações, os jogadores trocam informações e realizam a tomada de decisão (de acordo com as regras inicialmente estabelecidas) para os papéis escolhidos. Normalmente, a duração deste passo é definida no início do jogo (por exemplo, 10 minutos). Em alguns casos, é necessário um tempo maior para este passo, dependendo da quantidade de jogadores, dificuldade das regras do jogo, etc. Por exemplo, o industrial pode decidir aumentar a sua produção, vender propriedades, etc. O protocolo de negociação a ser utilizado é independente da metodologia, sendo a escolha dependente de como a mesma será implementada;
3. Os jogadores informam ao simulador multiagentes quais foram suas ações escolhidas;
4. Os dados são computados pelo simulador (processamento das ações): estas ações irão modificar o cenário inicial. As propriedades do ambiente são modificadas, o que implica na modificação de dados de cada papel. Por exemplo, se o jogador que representa o industrial resolve instalar uma nova indústria no cenário, o jogador que representa o ecologista perceberá a mudança no índice de poluição das águas. Este passo é o final do primeiro turno de jogo;
5. O simulador retorna o novo cenário, após as alterações realizadas pelas ações escolhidas pelos jogadores.

Um novo turno é então iniciado, retornando-se ao passo 2. Esta seqüência de passos poderá ser repetida diversas vezes, dependendo dos objetivos do jogo. Normal-

mente, o primeiro turno de simulação é mais longo (tempo de duração), pois os jogadores estão conhecendo melhor as regras e como manipular os recursos que o jogo oferece. Os turnos seguintes são mais curtos, pois os jogadores já têm um objetivo e definiram as estratégias para conquistá-lo. Independentemente do número de rodadas, ao final, sempre é realizado uma discussão (*debriefing*) sobre as escolhas feitas por cada jogador, com o objetivo de entender melhor a problemática e possíveis soluções apresentadas [Dorn 1989, Egenfeldt-Nielsen 2004].

## 5. Estudos de Caso: Protótipos JogoMan e ViP-JogoMan

O domínio de problema escolhido foi o da gestão de recursos naturais, mais especificamente a problemática dos recursos hídricos <sup>1</sup>. A gestão de recursos naturais é um tópico que se refere a grandes áreas (físicas) de trabalho e pesquisa, como cidades, estados, etc., onde todo o ecossistema deve ser analisado. Além das dimensões das regiões, ainda se tem uma grande diversidade de atores, com objetivos e estratégias diferentes. Por exemplo, o estudo dos problemas de água na Região Metropolitana de São Paulo é bastante complicado, pois trata-se de uma região que abrange aproximadamente 8.000 Km<sup>2</sup> de área, e mais de 1.500 Km<sup>2</sup> desta área são povoados por aproximadamente 18 milhões de habitantes. A problemática está na organização do processo de negociação entre estes atores, visto que, tendo objetivos e estratégias diferentes, normalmente ocorrem conflitos e o conhecimento não é comum entre estes atores. Para iniciar qualquer negociação, é necessário que os atores estejam "falando a mesma língua", ou seja, é necessário que exista um dicionário comum, com termos, definições, etc. Um bom exemplo é colocar um agricultor e um industrial conversando: eles não entrarão em consenso facilmente, pois seus objetivos geram conflito de interesses. Depois de iniciado o processo de negociação, as possíveis soluções podem divergir entre grupos de atores, ou seja, uma boa solução para alguns pode não ser para outros. Para auxiliar esse processo, é necessária a utilização de sistemas de apoio à decisão em grupos (SAD-G), que podem ajudar a solucionar a divergência de conhecimentos e conflitos entre as soluções.

No domínio da gestão de recursos naturais, tanto MABS quanto RPG têm sido utilizados para desenvolver SAD-Gs [Guyot and Honiden 2006, Barreteau et al. 2004, Etienne 2003]. Neste trabalho, o uso conjunto destas duas técnicas (RPG e MABS) é denominado GMABS. Em MABS, os agentes e as interações destes com o ambiente e entre si podem ser facilmente analisados, e na gestão de recursos naturais é a inter-relação entre os atores e o ambiente o foco de estudo da negociação. RPGs são muito utilizados para treinamento em diversas áreas e auxiliam muito na comunicação entre diferentes tipos de atores, fazendo com que temas importantes possam ser estudados e analisados de forma lúdica [Barreteau et al. 2001, Klimick and Bettocchi 2003].

### 5.1. Protótipo JogoMan

O protótipo **JogoMan** (Jogo dos Mananciais) tem por objetivo um melhor entendimento do domínio de gestão de recursos naturais, da metodologia GMABS e sua aplicação em SAD-Gs [Adamatti et al. 2005]. Este protótipo é uma simplificação da realidade e envolve problemas de gerenciamento de água e de solo em 3 diferentes cidades e o geren-

---

<sup>1</sup>Tal escolha se baseia no fato de que este trabalho se insere no contexto de um projeto nesta área, denominado Negowat Project: *Facilitating Negotiations Over Land and Water Conflicts in Latin American Peri-Urban Upstream Catchments: Combining Multi-Agent Modelling with Role-Playing Games*. Comunidade Européia: proc. número ICA4-CT-2002-10061.

ciamento da pressão urbana sobre esta região. Neste protótipo, existem quatro tipos de jogadores:

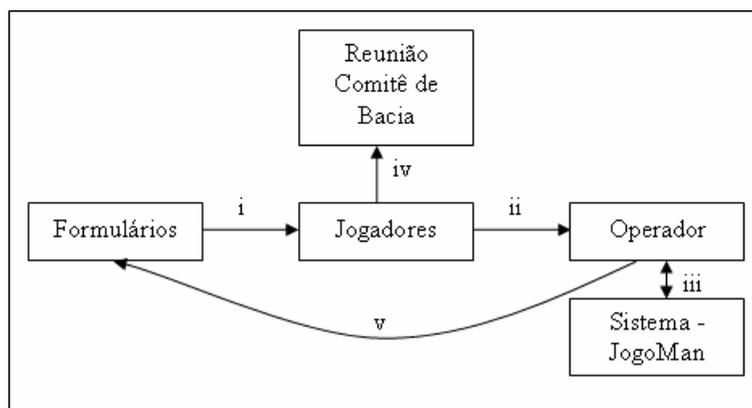
- *Prefeitos*: cada cidade tem uma atividade principal diferente: urbana, agrícola ou de áreas preservadas (floresta). Desta maneira, cada prefeito tem objetivos diferentes. Por exemplo, se a cidade C é uma área preservada, o jogador que assumir o papel de prefeito de C deve persistir no objetivo de manter o município preservado. Os prefeitos podem construir novas infra-estruturas nas cidades, como redes de água potável e/ou de saneamento ou construção de escolas e/ou hospitais. Por exemplo, durante o jogo, no processo de negociação (RPG), os proprietários particulares deveriam pedir, ou até mesmo exigir, que os prefeitos implementassem as infra-estruturas em suas cidades, visto que os proprietários pagam impostos;
- *Administrador da Companhia de Água (ÁguaPura)*: a companhia de água é uma empresa pública e pode implantar rede de água potável e de saneamento em qualquer área dos três municípios;
- *Proprietários Particulares*: cada um dos 9 proprietários possui 5 áreas (5 parcelas do espaço), com diferentes usos do solo, como floresta ou agricultura não-irrigada. Para cada tipo de uso do solo, há diferentes valores de manutenção (impostos) e retorno financeiro. Cada área pode ser vendida ou pode ter seu uso do solo modificado;
- *Representante dos Sem Teto*: este jogador tem uma função especial no jogo, pois é responsável pela alocação de um número determinado de novas famílias em cada ciclo de simulação. Estas novas famílias chegam à região (o que caracteriza a pressão urbana) e podem ser alocadas em loteamentos (com infra-estrutura) ou em favelas (sem infra-estrutura). Dependendo de como estas famílias forem alocadas, a qualidade e/ou quantidade de água da região é alterada.

Cada jogador escolhe suas ações individualmente. Contudo, este deve saber que suas ações podem influenciar todos os outros jogadores, porque a qualidade e quantidade de água dependem do uso do solo. Por exemplo, suponha que um dos prefeitos decida reduzir os impostos para que os proprietários particulares preservem suas áreas (florestas). Como resultado desta ação, vários proprietários particulares podem decidir manter suas áreas com florestas ou então investir em reflorestamento. Essa ação beneficia a qualidade e a quantidade da água. Em outro exemplo, caso um proprietário particular decida construir uma indústria, visando maior retorno financeiro, sua ação pode eventualmente prejudicar imensamente a qualidade das águas.

No JogoMan, os jogadores não têm acesso direto ao sistema computacional. Eles recebem formulários em papel em cada turno do jogo para preencher suas ações e ao final do turno, um operador insere as ações escolhidas no simulador MABS. Também ao final de cada turno de jogo uma reunião, denominada Reunião do Comitê de Bacia, é realizada, onde os jogadores definem estratégias coletivas a serem seguidas nos próximos turnos. A Figura 2 apresenta o funcionamento operacional do protótipo JogoMan.

### 5.1.1. Testes Realizados

Foram realizados diversos testes com o protótipo JogoMan. Primeiramente, foram realizados testes com membros do Projeto Negowat (pesquisadores e estudantes de graduação)



**Figura 2. Funcionamento operacional do JogoMan.**

para verificar se o protótipo estava funcional e correto. Depois, foram realizados testes com alunos de graduação e pós-graduação de quatro universidades diferentes da região de São Paulo. O cenário utilizado em todos os testes do protótipo envolvia 14 jogadores: 9 proprietários particulares, 3 prefeitos, 1 representante dos SemTeto e 1 administrador da AguaPura.

Para aplicação dos testes, a seguinte seqüência de passos foi utilizada:

1. Explicação geral a todos os jogadores sobre o jogo, apresentando os objetivos, regras e papéis existentes no jogo;
2. Cada jogador escolhe um dos papéis (tipo de jogador) e preenche o questionário pré-jogo<sup>2</sup>. Tanto o questionário pré-jogo quanto o pós-jogo utilizam perguntas semi-estruturadas e dissertativas;
3. Para cada tipo de jogador são dadas mais informações específicas do papel escolhido;
4. Inicia-se o primeiro turno de jogo e o tempo de negociação entre os jogadores e escolha das ações é definido (normalmente de 30 a 40 minutos);
5. Jogadores informam ao operador do simulador as ações escolhidas no turno;
6. As ações são computadas no MABS. Estas ações modificam o cenário inicial de jogo, e tem-se o final do primeiro turno;
7. Os jogadores se reúnem na Reunião do Comitê de Bacia para discutir as estratégias coletivas a serem tomadas no próximo turno de jogo. Normalmente, os jogadores no papel de prefeitos ou de administrador da AguaPura são cobrados em relação a infra-estruturas, impostos, etc., por parte dos proprietários particulares.
8. Ao final dos turnos de jogo (normalmente 3 ou 4 turnos) é realizada uma reunião entre todos os jogadores, chamada de *debriefing*. Nesta reunião final, os jogadores discutem dúvidas e problemas do protótipo e preenchem o questionário pós-jogo<sup>3</sup>. Com os dados desta reunião e dos questionários, é possível realizar uma avaliação qualitativa do protótipo.

Pelos dados obtidos pelos questionários pré e pós-jogo respondidos pelos jogadores, podem-se assinalar algumas sugestões/críticas:

<sup>2</sup>O questionário pré-jogo tem como objetivo descobrir o perfil dos jogadores, pois isso irá influenciar na forma que a análise dos testes será realizada.

<sup>3</sup>O questionário pós-jogo tem como objetivo verificar o que os jogadores entenderam sobre o domínio do jogo e a forma de execução do mesmo (metodologia GMABS).

- A maioria dos jogadores achou o jogo muito interessante e realística, ajudando a entender a realidade das bacias peri-urbanas;
- Os jogadores aprenderam sobre o domínio do problema do jogo, pois o RPG é uma forma didática e divertida de aprender um novo tópico;
- Os jogadores acharam que o jogo tinham uma grande quantidade de regras e sugeriram um manual mais completo para o jogo, para entendimento das regras e assim, melhores formas para realizarem a negociação;
- Os jogadores acharam que o tempo de cada turno, entre 30 a 40 minutos, muito curto para tomada de decisão, pois o jogo é muito complexo.

Cada um dos testes realizados com o protótipo JogoMan apresentou resultados diferentes, e isso é parte inerente da utilização da metodologia GMABS, visto que grupos diferentes de pessoas estão envolvidas em cada um dos jogos e estas escolhem diferentes ações em diferentes jogos.

Pode-se concluir que o protótipo JogoMan, que é uma instância de um SAD-G utilizando a metodologia GMABS, auxilia no entendimento do domínio do problema proposto, a gestão de recursos hídricos. Em particular, propicia o entendimento de quanto complexo é o processo de negociação neste domínio e que a definição de estratégias coletivas é algo extremamente difícil, já que há inúmeros conflitos de interesse. A avaliação utilizada neste protótipo foi baseada na aplicação de questionários e pelas respostas obtidas, pode-se verificar também que a forma proposta para interação, a partir de um jogo, é muito divertida e didática, fazendo com que os jogadores tenham maior interesse no problema e em interagir para encontrar possíveis soluções.

## **5.2. Protótipo ViP-JogoMan**

Os jogos do tipo RPG definidos para gestão de recursos naturais normalmente necessitam de um grande número de jogadores. Mesmo sendo o jogo uma abstração de um problema específico, este exige um número mínimo de jogadores e regras para que o grau de abstração não seja tão alto, de modo a oferecer um auxílio ao entendimento do processo de negociação real. Após a realização dos testes no protótipo JogoMan, verificou-se a necessidade de alguma ferramenta que possa substituir os jogadores reais, caso o número necessário não for suficiente, pois agendar uma sessão de jogo com um grupo de 14 pessoas (número típico de jogadores necessários para o protótipo JogoMan) é uma tarefa complicada. Desta maneira, surgiu a idéia de criação de Jogadores Virtuais, que além de substituir os jogadores reais, podem ajudar a forçar uma determinada situação. Por exemplo, para o domínio de gestão de recursos naturais, se os jogadores virtuais tiverem como objetivo a melhoria da qualidade da água, estes podem escolher não desmatar as florestas. Esta ação pode ser vista pelos outros jogadores, que perceberão os impactos desta ação sobre o cenário, ou seja, a melhoria da qualidade da água.

Assim, um novo protótipo foi desenvolvido, baseado no protótipo JogoMan, onde estão inseridos jogadores virtuais. Este protótipo chama-se ViP-JogoMan (*Virtual Players in JogoMan*) e tem como objetivo o estudo da inserção de jogadores virtuais na metodologia GMABS. ViP-JogoMan utiliza as mesmas regras e papéis do protótipo JogoMan. Contudo, neste novo protótipo os jogadores poderão estar em lugares remotos, pois o jogo é jogado via Web. Desta forma, os jogadores terão acesso ao jogo com o sistema computacional (via Web), mas não terão interação física entre si, pois poderão estar em lugares diversos.

No protótipo ViP-JogoMan, além de verificar se jogadores virtuais são verossímeis em suas tomadas de decisão, outro aspecto também influencia muito a interação entre os jogadores: a tecnologia. Os jogadores estarão remotamente localizados, e as negociações/interações entre os jogadores podem ocorrer de forma errônea. Assim, também é necessário verificar se a tecnologia influencia a utilização da metodologia GMABS.

### 5.2.1. Jogadores Virtuais

O aspecto mais importante para o desenvolvimento de ViP-JogoMan foi a definição dos Jogadores Virtuais. A arquitetura BDI (*Beliefs, Desire and Intentions*) [Rao 1996] foi escolhida para tomada de decisão dos jogadores. Esta arquitetura utiliza uma base de conhecimento baseada em crenças, desejos e intenções para os jogadores.

Para definir a base de conhecimento dos jogadores, todas ações escolhidas pelos jogadores nos testes realizados no protótipo JogoMan foram mapeadas para verificar possíveis estratégias existentes, a partir de seqüências de ações que se repetiam em diferentes jogos. Por exemplo, o papel de administrador da AguaPura pode implantar rede de água potável e de esgoto nos lotes, sem haver nenhuma restrição entre as duas infraestruturas. Contudo, em todos os jogos no protótipo JogoMan, todos os jogadores que desempenharam este papel, apenas instalaram a rede de esgoto onde já existisse rede de água potável. Isso confirma que existe a formação autônoma de estratégias de jogo para cada tipo de jogador, a partir das regras pré-definidas do jogo. Foi percebido que as estratégias encontradas para cada tipo de jogador (Proprietários Particulares, Prefeitos, AguaPura e Sem Teto) possuíam diferentes interesses, algumas com maior preocupação econômica, outras com maior preocupação ambiental, e ainda outras com maior preocupação social. A partir do interesse de cada estratégia, ou seja, seus objetivos, foram definidos perfis comportamentais para os jogadores. A Tabela 1 apresenta os nove perfis comportamentais definidos para os tipos de papéis existentes.

Jogador	Perfil Comportamental	Objetivo
Proprietário Particular	Econômico Ambientalista	Retorno financeiro Preocupação ambiental
Adm. AguaPura	Racional Protetor-Ambientalista	Preocupação social e lucratividade Preocupação Social
Prefeito	Social Ambientalista Econômico	Preocupação Social Preocupação ambiental Retorno financeiro
Rep. SemTeto	Econômico Social	Retorno financeiro Preocupação social

**Tabela 1. Perfis Comportamentais Definidos.**

Por exemplo, o jogador Proprietário Particular com o perfil comportamental com interesse econômico adota as seguintes estratégias:

- Se tiver áreas próximas as áreas urbanas, trocar do uso do solo para Loteamento, para comercialização junto ao Representante dos SemTeto. Isto porque o Representante dos SemTeto sempre dá preferência por áreas próximas as áreas urbanas;

- Tenta vender áreas com Loteamento ao Representante dos SemTeto. O valor de oferta mínimo é o retorno do investimento realizado (implantação do Loteamento);
- Se o uso do solo atual não for Floresta, trocar de uso do solo para Agricultura e/ou Agricultura Irrigada, por ser um uso do solo que o investimento é baixo e tem retorno rápido, comparado aos outros usos do solo, como Indústria;
- Se o uso do solo atual for Floresta, trocar para Plantação, visando o ganho de supressão do corte das árvores. Depois, utilizar a estratégia anterior, transformando em Agricultura/Agricultura Irrigada.

### 5.2.2. Testes Realizados

Foram realizadas três sessões de jogos com o protótipo ViP-JogoMan, envolvendo pessoas e jogadores virtuais, todos via Web. Estes testes tinham como objetivo verificar a influência da tecnologia na metodologia GMABS e se a utilização de perfis comportamentais nos jogadores virtuais foi apropriada para a tomada de decisão deste tipo de jogador.

Para aplicação dos testes, a seguinte seqüência de passos foi utilizada:

1. Alguns dias antes da realização do jogo, os jogadores recebem por *email* um manual de instruções com as explicações gerais do jogo, objetivos, regras e papéis;
2. Cada jogador escolhe um dos papéis disponíveis (fora os papéis anteriormente escolhidos para os jogadores virtuais);
3. Para cada tipo de jogador, mais informações específicas são repassadas via *email*, juntamente com um questionário pré-jogo;
4. Um link para acesso as interfaces do jogo fica disponível (on-line), com intuito de familiarizar os jogadores, antes do início do jogo;
5. No dia e na hora marcados para realização do jogo, todos os jogadores acessam o sistema e interagem via interface gráfica com os outros jogadores e escolhem suas ações para cada turno. Ao final do tempo do turno, os jogadores devem informar ao sistema suas ações. Finaliza-se o primeiro turno de jogo;
6. Todos os jogadores vão para uma interface única, onde acontece a Reunião do Comitê de Bacia, para definir as estratégias coletivas para os próximos turnos de jogo;
7. Ao final do quarto turno de jogo, todos os jogadores recebem gráficos explicativos sobre a situação global do sistema, com valores de índice de poluição do reservatório, percentual de famílias conectadas a rede de água e de esgoto, percentual de famílias morando em favelas, para entendimento global do sistema;
8. Os jogadores devem enviar por *email* um questionário pós-jogo.

O protótipo ViP-JogoMan, por ser jogado via Web permite o armazenamento de todas as negociações realizadas durante os turnos do jogo. Segundo McKersie e Fonstad [McKersie and Fonstad 1997], em negociações realizadas via Internet, tudo é armazenado e assim é possível analisar os dados com bastante atenção e entender melhor todo o processo de negociação. No protótipo JogoMan, as únicas informações dos jogadores eram suas escolhas finais em negociações, anotadas nos formulários em papel entregues ao operador do sistema. Já no protótipo ViP-JogoMan pode-se acompanhar todas as interações realizadas pelos jogadores durante o processo de negociação, inclusive

negociações não concluídas ou rejeitadas. Além da análise das negociações dos jogadores, ainda foram aplicados questionários pré e pós-jogo a todos os jogadores reais (pessoas). Estes questionários tinham como objetivo verificar a influência da tecnologia no processo de negociação e a percepção dos jogadores reais aos jogadores virtuais.

Pelas respostas dos jogadores nos questionários pré e pós-jogo, pode-se concluir que o protótipo ViP-JogoMan ofereceu aos jogadores sensação de interação, divertimento e aprendizado. Também conclui-se que os jogadores virtuais tiveram tomada de decisão parecida aos dos jogadores reais, pois a maioria dos jogadores reais não percebeu que estava jogando com um jogador virtual e quando percebeu foi pela velocidade das respostas dadas, não pela forma como interagiram com os outros jogadores.

Já pela análise de todas as interações dos jogadores durante o processo de negociação, pode-se concluir que todos os jogadores conseguiram negociar de forma ampla, visto pela grande quantidade de interações entre os jogadores durante os turnos de jogo. A Tabela 2 apresenta uma análise quantitativa de negociações concluídas entre os jogadores para a ação "compra" de lotes para três sessões de testes realizadas nos dois protótipos. Entende-se por negociações concluídas aquelas que realmente geram ações. No caso da ação compra, uma negociação é dita concluída quando o jogador que possui um lote aceita vender por determinado preço este lote a outro jogador. No protótipo JogoMan, apenas as ações concluídas foram armazenadas durante o jogo, visto que os jogadores as anotavam nos formulários a serem entregues ao operador para inserção dos dados no MABS. Desta maneira, somente a comparação entre negociações concluídas é possível de ser realizada entre os dois protótipos. Pela Tabela 2, verifica-se que o protótipo ViP-JogoMan apresenta um número de negociações concluídas muito maior que o protótipo JogoMan. Uma possível explicação para isso se deva ao fato de que, segundo Peppet [Peppet 2002], em negociações via Internet, as pessoas sentem-se mais à vontade em expressar suas opiniões a distância, sendo assim mais produtivas. Outra explicação pode ser que, em jogos presenciais, como o protótipo JogoMan, os jogadores interagem (conversam) muito, e algumas vezes perdem o foco de negociação, concluindo menos negociações por turno de jogo [Van Es et al. 2004].

	JogoMan	ViP-JogoMan
Sessão teste 1	2	17
Sessão teste 2	2	14
Sessão teste 3	6	12

**Tabela 2. Negociações concluídas entre os jogadores para a ação compra de lotes.**

Pelos dados obtidos pelos questionários pré e pós-jogo respondidos pelos jogadores, algumas melhorias/críticas foram apresentadas pelos jogadores:

- Apresentar os gráficos da evolução global do sistema ao final de cada uma das rodadas do jogo, para que a evolução fique mais clara no decorrer do jogo;
- Identificar cada jogador por uma foto (figura) para facilitar a negociação;
- Melhorar a interface gráfica, no que tange ao envio e ao recebimento de mensagens, para que seja mais intuitiva. Um dos jogadores sugeriu um sistema de cores, onde cada jogador tem uma cor e assim fica mais fácil identificar as mensagens recebidas de cada jogador.

## 6. Conclusões e Trabalhos Futuros

A metodologia GMABS pode ser utilizada com Sistema de Apoio à Decisão em Grupos (SAD-Gs), auxiliando na negociação e na resolução de conflitos em diversos domínios, como na gestão de recursos naturais, pois une a capacidade dinâmica de MABS com a capacidade de discussão e aprendizado das técnicas de RPG [Barreteau et al. 2001]. Isso pôde ser verificado pelos testes realizados nos protótipos JogoMan e ViP-JogoMan.

Também pode-se concluir, pelos testes no protótipo ViP-JogoMan, que a utilização da metodologia GMABS via *Web* é eficiente e prática, pois disponibiliza sua utilização de lugares remotos, viabilizando sua utilização por um grupo maior de pessoas. Sendo um dos objetivos dos testes no protótipo ViP-JogoMan verificar se a tecnologia (computador) influencia na interação entre os jogadores, pode-se concluir que o computador auxilia na maior utilização deste tipo de jogo de duas formas:

- Disponibiliza o jogo de forma remota, facilitando a participação dos jogadores;
- Possibilita o mapeamento completo de todo o processo de negociação entre os jogadores, sendo possível analisar as negociações rejeitadas e/ou não concluídas.

Contudo, não pode-se afirmar que o número de negociações seja diretamente ligado ao aprendizado do processo de negociação. Muitos jogadores nos testes do protótipo JogoMan afirmaram ter entendido a problemática de negociação existente, pois cada jogador tinha um foco de negociação diferente, devido a seus objetivos no jogo. Muitos jogadores do protótipo ViP-JogoMan também afirmaram que o processo de negociação exige muita comunicação e interação para ocorrer de forma ampla. Desta maneira, os dois protótipos atenderam ao objetivo principal ao que foram propostos: auxiliar no entendimento do processo de negociação para o domínio de problema exposto. Sendo estes protótipos instâncias de SAD-Gs, espera-se que com o entendimento do processo de negociação, estes jogadores possam interagir em situações reais de maneira mais eficiente.

Outro aspecto analisado foi a utilização de jogadores virtuais na metodologia GMABS. Pode-se concluir que este tipo de jogador, implementado a partir de perfis comportamentais e da arquitetura BDI, teve tomada de decisão não trivial perante os outros jogadores, sendo uma ferramenta útil no preenchimento de jogadores para atingir o número necessário, ou para apresentar novas visões de jogo aos outros jogadores.

Nos testes realizados nos dois protótipos, algumas sugestões de melhoria foram apresentadas pelos jogadores. Em relação ao protótipo JogoMan, as principais críticas estavam relacionadas à quantidade de regras do jogo. Assim, quando da aplicação do protótipo ViP-JogoMan, tentou-se suprir este problema entregando a todos os jogadores manuais do jogo mais completos, com um número maior de informações sobre as regras gerais do jogo, bem como de cada tipo de jogador. Já nos testes do protótipo ViP-JogoMan, as principais críticas estavam relacionadas à interface de gráfica para envio e recebimento de mensagens. Um trabalho futuro muito importante é a definição e implementação de uma interface gráfica para troca de mensagens mais intuitiva.

Outro trabalho futuro, que poderá melhorar a tomada de decisão dos jogadores virtuais, é o desenvolvimento de uma base de conhecimento dinâmica, onde os jogadores incrementem suas estratégias durante o jogo, pois a implementação atual destes jogadores é estática. Contudo, jogadores reais, durante o jogo, vão aprendendo e incrementando suas tomadas de decisão. Assim, implementar uma base de conhecimento dinâmica nos jo-

gadores virtuais, através de técnicas de aprendizagem de máquina, tornaria suas tomadas de decisão mais realistas.

## Referências

- Adamatti, D., Sichman, J., Bommel, P., Ducrot, R., Rabak, C., and Camargo, M. (2005). JogoMan: A prototype using multi-agent-based simulation and role-playing games in water management. In Ferrand, N., editor, *Join Conference on Multi-Agent Modeling for Environmental Management. CABM-HEMA-SMAGET*, Bourg-Saint-Maurice, Les Arcs, France. IRD.
- Alvares, L. O. C. and Sichman, J. S. (1997). Introdução aos sistemas multiagentes. In *Jornada De Atualização Em Informática*, pages 1–37, Brasília - UnB. Sociedade Brasileira de Computação.
- Barreteau, O., Bousquet, F., and Attonaty, J. (2001). Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to Senegal River Valley irrigated systems. *JASSS*, 4(2). <http://www.soc.surrey.ac.uk/JASSS/4/2/5.html>.
- Barreteau, O., Bousquet, F., Millier, C., and Weber, J. (2004). Suitability of multi-agent simulations to study irrigated system viability: Application to case studies in the segal river valley. *Agricultural Systems*, 80(3):255–275.
- Barreteau, O., Le Page, C., and D'Aquino, P. (2003). Role-playing games, models and negotiation. *JASSS*, 6(2). <http://jasss.soc.surrey.ac.uk/6/2/10.html>.
- D'Aquino, P., Le Page, C., Bousquet, F., and Bah, A. (2003). Using self-designed role-playing games and a multi-agent systems to empower a local decision-making process for land use management: The selfcormas experiment in Senegal. *JASSS*, 6(3). <http://jasss.soc.surrey.ac.uk/6/3/5.html>.
- Dare, W. and Barreteau, O. (2004). Agent-based simulatins backing use of role-playing games as dialogue support tools: Teaching from experiments. In Gilbert, N. and Troitzsch, K. G., editors, *Sixth International Conference on Social Science Methodology*, Amsterdam. Social Science Methodology.
- Doran, J. (2001). Intervening to achieve co-operative ecosystem management: Towards an agent based model. *JASSS*, 4(2). <http://www.soc.surrey.ac.uk/JASSS/4/2/4.html>.
- Dorn, D. S. (1989). Simulation games: One more tool on the pedagogical shelf. *Teaching Sociology*, 17(1):1–18.
- Dornelas, J. S. (2000). *Impactos da adoção de sistemas de apoio à decisão para grupos em um processo decisório público participativo: o caso do orçamento de Porto Alegre*. Doutor em administração, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- Drogoul, A. and Ferber, J. (1992). Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In *Proceedings of MAAMAW'92*, Viterbo. Elsevier North-Holland.
- Egenfeldt-Nielsen, S. (2004). Review of the research on educational usage of games. <http://itu.dk/people/sen/public.htm>.

- Etienne, M. (2003). SYLVOPAST: a multiple target role-playing game to assess negotiation processes in sylvopastoral management planning. *JASSS*, 6(2). <http://jasss.soc.surrey.ac.uk/6/2/5.html>.
- Frozza, R. (1997). SIMULA: Ambiente para desenvolvimento de sistemas multiagentes reativos. Master's thesis, UFRGS, Universidade Federal do Rio Grande do Sul - Brasil, Rio Grande do Sul, Brasil.
- Gilbert, N. and Troitzsch, K. G. (1999). *Simulation for the Social Scientist*. Buckingham and Philadelphia: Open University Press, London.
- Guyot, P. and Honiden, S. (2006). Agent-based participatory simulations: Merging multi-agent systems and role-playing games. *Journal of Artificial Societies and Social Simulation*, 9(4). <http://jasss.soc.surrey.ac.uk/9/4/8.html>.
- Klimick, C. (2003). Construção de personagem & aquisição de linguagem: O desafio do RPG no INES. Dissertação de mestrado, Programa de Pós-graduação do Departamento de Artes e Design - PUC, Rio de Janeiro.
- Klimick, C. and Bettocchi, E. (2003). O lugar do virtual no RPG, o lugar do RPG no design. In *II Simpósio do LARS: Atopia*, Departamento de Artes e Design, PUC-Rio. <http://www.historias.interativas.nom.br/historias/textos/rpgvirtual.htm>.
- Marietto, M. G. B., David, N., Sichman, J. S., and Coelho, H. (2002). Requirements analysis of multi-agent based simulation platforms. Technical Report LTI 01/2002, University of São Paulo, Universidade de São Paulo - São Paulo.
- McKersie, R. B. and Fonstad, N. O. (1997). Teaching negotiation theory and skills over the internet. *Negotiation Journal*, 13(4):363–368.
- Peppet, S. R. (2002). Teaching negotiation using web-based streaming video. *Negotiation Journal*, 18(3):271–283.
- Rao, A. S. (1996). AgentSpeak (L): BDI agents speak out in a logical computable language. In de Velde, W. V. and Perram, J., editors, *Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, pages 42–55, London. Eindhoven - The Netherlands, Lecture Notes in Artificial Intelligence - Springer-Verlag.
- Strack, J. (1984). *GPSS: modelagem e simulação de sistemas*. LTC, Rio de Janeiro. 174 p.
- Van Es, R., French, W., and Stellmazel, F. (2004). Resolving conflicts over ethical issues: Face-to-face versus internet negotiations. *Journal of Business Ethics*, 53(1):165–172.

# WebBEMS: um sistema baseado em componentes para o suporte à colaboração via Web

Cléver R. Guareis de Farias<sup>1</sup>, Carlos E. Gonçalves<sup>2</sup>, Marta Costa Rosatelli<sup>2</sup>

<sup>1</sup>Departamento de Física e Matemática  
Universidade de São Paulo (FFCLRP/USP)  
Av. Bandeirantes, 3900 – CEP 14040-901 – Ribeirão Preto – SP

<sup>2</sup>Programa de Mestrado em Informática  
Universidade Católica de Santos (Unisantos)  
R. Dr. Carvalho de Mendonça, 144 – CEP 11070-906 – Santos – SP

farias@ffclrp.usp.br, ceg-elus@unisantos.br, rosatelli@unisantos.br

**Abstract.** *Component-based development has been increasingly used in the development of collaborative services and systems. Software components allow solving several problems related to the distribution of collaborative systems. Besides, components can be configured, replaced and combined dynamically, increasing the flexibility, integration and adequacy of collaborative systems. This paper presents the development of an electronic meeting system named Web-Based Electronic Meeting System (WebBEMS), according to an architectural model conceived for the development of component-based collaborative systems. The paper depicts details of modelling, implementation and test of the application starting from the components architecture until its user interface.*

**Resumo.** *O desenvolvimento baseado em componentes tem sido cada vez mais utilizado no desenvolvimento de serviços e sistemas colaborativos. Os componentes de software possibilitam a resolução de vários problemas relacionados à distribuição dos sistemas colaborativos. Além disso, os componentes podem ser configurados, substituídos e combinados dinamicamente, aumentando dessa forma a flexibilidade, integração e adequação dos sistemas colaborativos. Este artigo apresenta o desenvolvimento de um sistema de reunião eletrônica, o Web-Based Electronic Meeting System (WebBEMS), segundo um modelo arquitetônico concebido para o desenvolvimento baseado em componentes de sistemas colaborativos. O artigo descreve em detalhes a modelagem, implementação e teste da aplicação partindo da arquitetura de componentes até a sua interface com o usuário.*

## 1. Introdução

O desenvolvimento baseado em componentes surgiu como paradigma de desenvolvimento de software no final dos anos 90. Este paradigma tem por princípio a construção de sistemas através da composição dos serviços individuais de um conjunto de componentes de software. Um componente de software é um pedaço binário de

software, auto contido, customizável e componível, com interfaces e dependências bem definidas [de Farias 2002].

O desenvolvimento baseado em componentes apresenta algumas vantagens se comparado ao desenvolvimento de software tradicional, tais como desenvolvimento das diversas partes que compõem um sistema de forma autônoma umas das outras, facilidade de instalação e manutenção de um sistema de software, e redução do tempo de desenvolvimento do sistema através do reuso.

Outra característica importante dos componentes de software é que os mesmos devem atender a uma especificação, que lhes conferem características importantes como compatibilidade e reusabilidade. Com a popularização dos componentes de software, surgiram várias especificações, desde as voltadas à componentização de interfaces gráficas, tais como as especificações *ActiveX*, *COM*, *VCL* e *JavaBeans*, até especificações mais complexas capazes de operar em um modelo de componentes distribuídos, tais como as especificações *CORBA*, *.NET* e *EJB*.

Componentes de software têm sido cada vez mais utilizados no desenvolvimento de serviços e sistemas colaborativos, c.f., [ter Hofte 1998, Brusilovsky 2004, Fuks *et al.* 2005, Pukkhem e Vatanawood 2005, de Farias *et al.* 2006]. Componentes são instalados e executados em plataformas distribuídas, contribuindo dessa forma para a resolução de vários problemas relacionados à distribuição dos sistemas colaborativos. Componentes também podem ser configurados, substituídos e combinados dinamicamente, aumentando dessa forma a flexibilidade, integração e adequação desses sistemas.

Um dos maiores desafios do desenvolvimento de um sistema baseado em componentes é a definição de sua arquitetura. O uso de um modelo arquitetônico concebido para o desenvolvimento baseado em componentes de sistemas colaborativos facilita não somente a definição da arquitetura deste sistema, mas também sua implementação e reuso dos componentes identificados. Neste sentido, este artigo apresenta o desenvolvimento de um sistema de reunião eletrônica, o sistema Web-Based Electronic Meeting System (WebBEMS), com base em um modelo arquitetônico projetado especificamente para atender sistemas colaborativos [de Farias 2005].

O restante deste artigo está estruturado da seguinte forma: a seção 2 apresenta uma visão geral do modelo arquitetônico utilizado neste trabalho; a seção 3 descreve os requisitos do WebBEMS, e detalhes da modelagem e arquitetura, mapeada para a tecnologia J2EE; a seção 4 descreve os detalhes de implementação e teste do WebBEMS; a seção 5 discute alguns trabalhos relacionados; finalmente na seção 6 são apresentadas algumas considerações finais e perspectivas futuras.

## **2. Modelo Arquitetônico**

A arquitetura de um sistema computacional pode ser definida como a estrutura (ou estruturas) do sistema em termos de um conjunto de componentes, as partes visíveis desses componentes e seus relacionamentos [Bass *et al.* 1997]. Neste sentido, a arquitetura de um sistema pode ser vista como a decomposição de mais alto nível deste sistema em um conjunto de componentes, conjuntamente com uma caracterização de como estes componentes interagem [van Vliet 2000]. Um modelo arquitetônico define

os componentes básicos utilizados na construção de uma aplicação e o relacionamento entre estes componentes.

Nosso modelo arquitetônico define diferentes tipos de componentes que servem como base para o refinamento do sistema e dos componentes desse sistema. O modelo também define diferentes tipos de aspectos colaborativos que auxiliam na identificação dos diferentes tipos de componentes e de seus serviços [de Farias *et al.* 2005, de Farias *et al.* 2006]. Três diferentes tipos de componentes são identificados: *componente de aplicação*, *componente colaborativo* e *componente básicos*.

Um componente de aplicação corresponde a uma aplicação colaborativa, a qual pode ser utilizada separadamente ou integrada junto a outros componentes de aplicação. Um componente de aplicação consiste de um conjunto integrado de componentes colaborativos. Um componente colaborativo representa um dado conceito ou aspecto colaborativo, o qual pode ser reutilizado na construção de outros componentes colaborativos mais especializados ou diretamente na construção de componentes de aplicação. Um componente colaborativo consiste de um conjunto integrado de componentes básicos. Um componente básico é a unidade mais básica de projeto, implementação e implantação. Seu comportamento é o resultado de um único conjunto de código binário. Componentes básicos os são utilizados para tratar da distribuição dos aspectos e responsabilidades colaborativos de um componente colaborativo.

Os diferentes aspectos funcionais de um componente colaborativo podem ser agrupados em diferentes camadas, chamadas de camadas de aspectos colaborativos. Foram identificadas quatro diferentes camadas: *interface*, *usuário*, *colaboração* e *recurso*. Cada camada faz uso da funcionalidade provida por uma camada inferior para prover alguma funcionalidade utilizada por uma camada superior. Cada camada pode ser implementada por um ou mais componentes básicos.

A camada de interface está relacionada com a provisão de uma interface adequada entre um usuário humano e o componente colaborativo. Esta camada trata de toda comunicação direta com os usuários. A camada de usuário está relacionada com o suporte local às atividades executadas por um único usuário. Esta camada trata de questões locais relacionadas a cada usuário individualmente, não afetando a colaboração como um todo. A camada de usuário também relaciona a camada de interface com a camada de colaboração. A camada de colaboração está relacionada com a lógica de colaboração de múltiplos usuários. Esta camada é responsável pela implementação dos principais aspectos da colaboração e pela relação da camada do usuário com a camada de recurso. A camada de recurso está relacionada com o acesso às informações colaborativas compartilhadas, as quais são mantidas de forma persistente em um banco de dados, por exemplo.

Os aspectos de colaboração de uma camada podem estar ou não acoplados. O conceito de *acoplamento* foi introduzido como um mecanismo geral para unir as contribuições de um conjunto de usuários, de tal forma que estes poderiam compartilhar a mesma visão ou estado de uma colaboração [ter Hofte 1998].

Uma camada está acoplada se todos seus usuários possuem a mesma percepção das informações presentes na camada e de alterações nas mesmas. Quatro níveis de acoplamento podem ser estabelecidos baseando-se nas camadas definidas neste trabalho:

*acoplamento de interface, acoplamento de usuário, acoplamento de colaboração e acoplamento de recurso.*

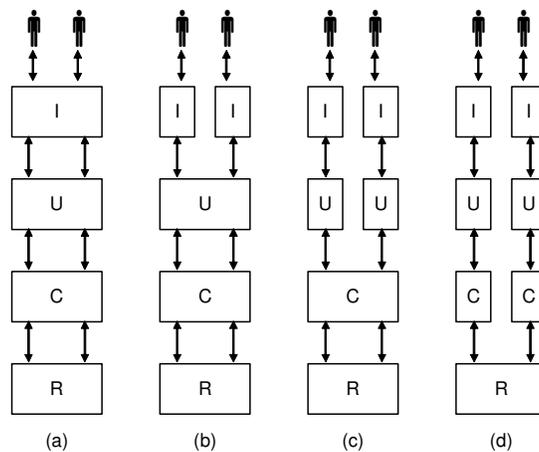
O nível de acoplamento de interface representa o nível de acoplamento mais restrito. Todos os usuários do componente têm a mesma percepção da colaboração a partir da camada de interface. Este nível de acoplamento corresponde ao estilo de colaboração conhecido como What You See Is What I See (WYSIWIS).

O nível de acoplamento de usuário oferece mais liberdade para os usuários de um componente do que o nível de acoplamento de interface. Todos os usuários do componente possuem a mesma percepção da colaboração a partir da camada do usuário, i.e., as informações na camada de usuário são compartilhadas por todos os usuários, mas suas camadas de interface são mantidas independentemente umas das outras.

O nível de acoplamento de colaboração oferece um grau maior de liberdade do que o nível de acoplamento de usuário. Todos os usuários do componente têm a mesma percepção da colaboração a partir da camada de colaboração, i.e., as informações na camada de colaboração são compartilhadas por todos os usuários, mas suas camadas de interface e usuário são mantidas independentemente umas das outras.

O nível de acoplamento de recurso provê o maior grau de liberdade para a utilização do componente. Neste nível, todos os usuários do componente têm a mesma percepção da colaboração a partir da camada de recurso, i.e., as informações na camada de recurso são compartilhadas por todos os usuários, mas suas camadas de interface, usuário e colaboração são mantidas independentemente umas das outras.

A Figura 1 ilustra os diferentes níveis de acoplamento definidos neste trabalho para dois usuários. As figuras 1a, 1b, 1c e 1d mostram os níveis de acoplamento de interface, de usuário, de colaboração e de recurso, respectivamente.



**Figura 1. Níveis de acoplamento**

### **3. Projeto do WebBEMS**

#### **3.1. Metodologia de desenvolvimento**

O projeto utilizou uma metodologia baseada em um ciclo de desenvolvimento incremental e iterativo, fazendo uso da linguagem UML [OMG 2007a, OMG 2007b] para a modelagem do sistema. Em linhas gerais uma metodologia de ciclo incremental e iterativo propõe o desenvolvimento de sistemas de software através de iterações encadeadas, onde em cada iteração uma nova versão do sistema é produzida a partir do resultado obtido na iteração anterior acrescido das novas demandas de requisitos para a iteração atual.

Uma iteração ou ciclo de desenvolvimento é composto por diversas atividades de projeto. Em cada atividade de projeto um grupo de tarefas bem definidas deve ser executado com a finalidade de produzir e/ou alterar um ou mais artefatos, sejam estes documentos, modelos ou códigos. Uma metodologia de ciclo incremental e iterativo normalmente é composta por quatro atividades básicas: especificação de requisitos, análise, implementação e teste.

O projeto do WebBEMS também aplicou algumas práticas descritas pela metodologia *Extreme Programming* [Jeffries *et al.* 2000], referentes a temas como planejamento de versões, testes de unidade, testes de aceitação e integração contínua.

#### **3.2. Requisitos do Sistema**

Em linhas gerais o sistema deve permitir que qualquer pessoa possa se cadastrar e criar reuniões eletrônicas compostas por uma sala de bate-papo em tempo real e por uma ou mais votações. Uma reunião possui apenas uma sala de bate-papo, a qual oferece recursos básicos para a formatação das mensagens e identificação dos seus autores.

O usuário responsável pela criação da reunião, chamado de coordenador, pode criar e fechar votações sempre que necessário. O coordenador também determina quando o resultado de uma votação estará disponível aos participantes, o que pode ocorrer antes ou depois de seu encerramento pelo coordenador. O coordenador também pode convidar outros usuários para participar da reunião através de convites. Um usuário convidado poder aceitar ou não um convite para participar de uma reunião, operação que deve ficar registrada e ser encaminhada na forma de confirmação para o coordenador da reunião.

Ao aceitar um convite o usuário torna-se um participante credenciado para atuar em uma reunião. Sempre que um participante entra ou sai de uma reunião os demais participantes são notificados. Os usuários que não aceitaram um convite para uma determinada reunião uma vez, não deverão mais receber convites para aquela reunião, evitando práticas inadequadas de insistência no envio do convite.

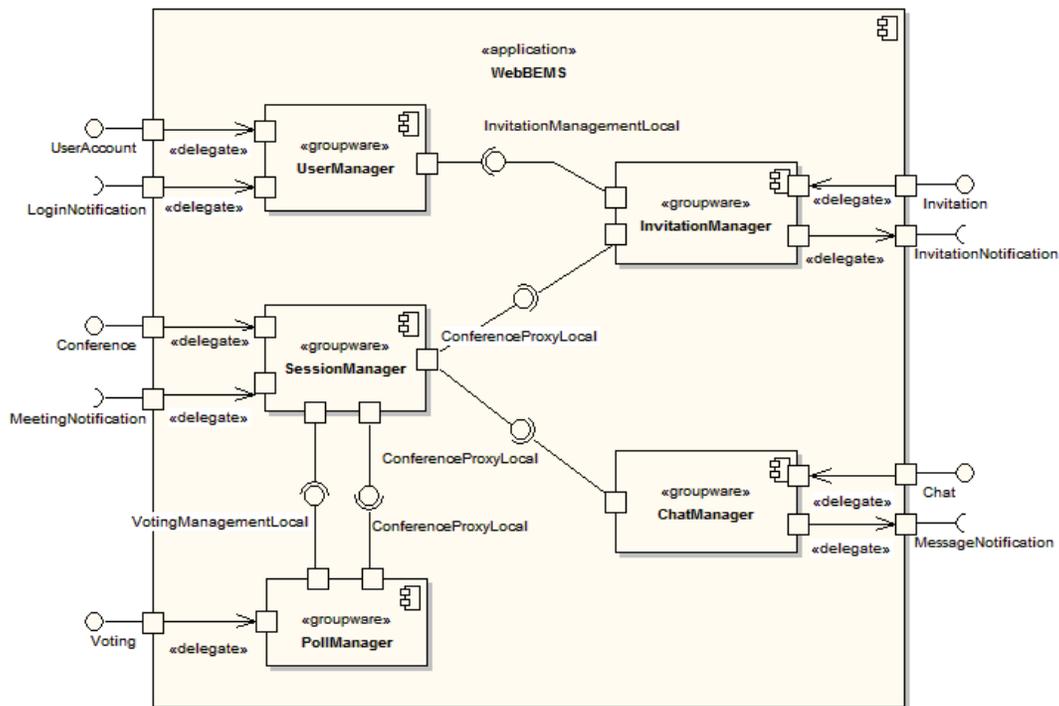
O coordenador pode remover um participante de uma reunião sempre que desejar, desde que o participante não esteja atuando na reunião naquele momento. Os participantes que forem removidos podem ser convidados novamente para a reunião da qual foram removidos.

Os requisitos não funcionais desejáveis para o sistema compreendem diversos aspectos, muitos dos quais são contemplados pela própria plataforma J2EE [Sriganesh

*et al.* 2006] escolhida para a implementação do sistema, tais como: a atomicidade das operações dentro do sistema; a independência do banco de dados usado pela aplicação para a persistência das informações; e a utilização de formas, cores, ícones e elementos que mantenham a interação dos usuários com o sistema fácil e agradável. Outros requisitos importantes como portabilidade, escalabilidade e desempenho são intrínsecos à plataforma J2EE e conseqüentemente presentes no WebBEMS.

### 3.3. Aspectos de Projeto

Com base nos requisitos funcionais o sistema foi dividido em 5 componentes gerenciadores: *UserManager*, *InvitationManager*, *SessionManager*, *PoolManager* e *ChatManager*. A Figura 2 apresenta a arquitetura do WebBEMS utilizando diagrama de componentes de UML. Os estereótipos <<application>>, <<groupware>> e <<basic>> foram utilizados para representar componentes de aplicação, colaborativo e básico, respectivamente.



**Figura 2. Arquitetura do WebBEMS**

Cada um destes gerenciadores é composto por um conjunto de componentes interdependentes estruturados de acordo com o nosso modelo arquitetônico [de Farias *et al.* 2005]. Estes componentes implementam as camadas de usuário, colaboração e recursos deste modelo.

As camadas de interface dos componentes gerenciadores foram agrupadas e implementadas no componente *Interface* do WebBEMS (veja Figura 4). Este componente é composto por um conjunto de objetos que permitem acessar as funções dos demais componentes dentro do sistema. Cada operação solicitada pelo usuário é processada por um ou mais objetos e estes se encarregam de se comunicar com os

componentes do sistema. Dessa forma, o cliente só visualiza e interage com as telas propagadas pelo servidor.

Os componentes da camada de usuário com interfaces assíncronas foram modelados como componentes do tipo *Message Driven Bean* da plataforma J2EE. Os demais componentes da camada do usuário, juntamente com os componentes da camada de colaboração, foram modelados como componentes do tipo *Stateless Session Bean* da plataforma J2EE. Os componentes da camada de recursos foram modelados como componentes do tipo *Entity Bean* da plataforma J2EE.

A Figura 3 apresenta a estrutura interna do componente *InvitationManager*. O componente *InvitationNotifier* foi modelado como um *Message Driven Bean*. Os componentes *InvitationProxy* e *InvitationManagement* foram modelados como *Stateless Session Beans*. O componente *Invitation* foi modelado como *Entity Bean*.

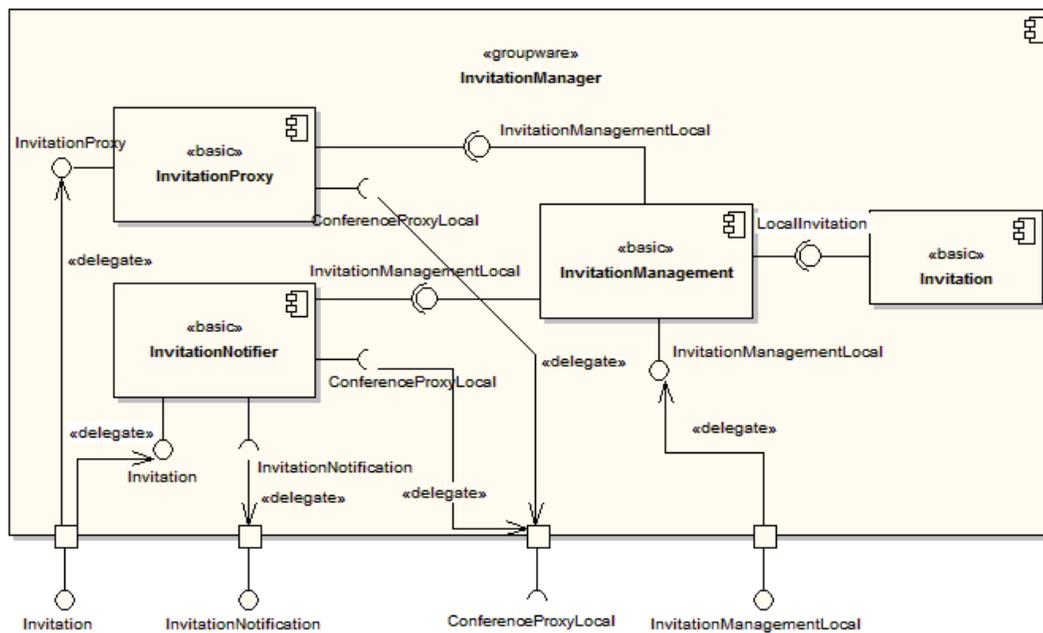


Figura 3. Componente *InvitationManager*

### 3.4. Aspectos de Interface e Comunicação

A interface Web foi desenhada para operar a partir das tecnologias J2EE *Servlet* e *JavaServer Pages* [Falkner e Jones 2003], em conjunto com o *framework Jakarta Struts* [Hightower 2004], que atribuem à aplicação outros padrões de projeto como *Model-View-Control*, *Front-Controller* e *Intercepting Filter*.

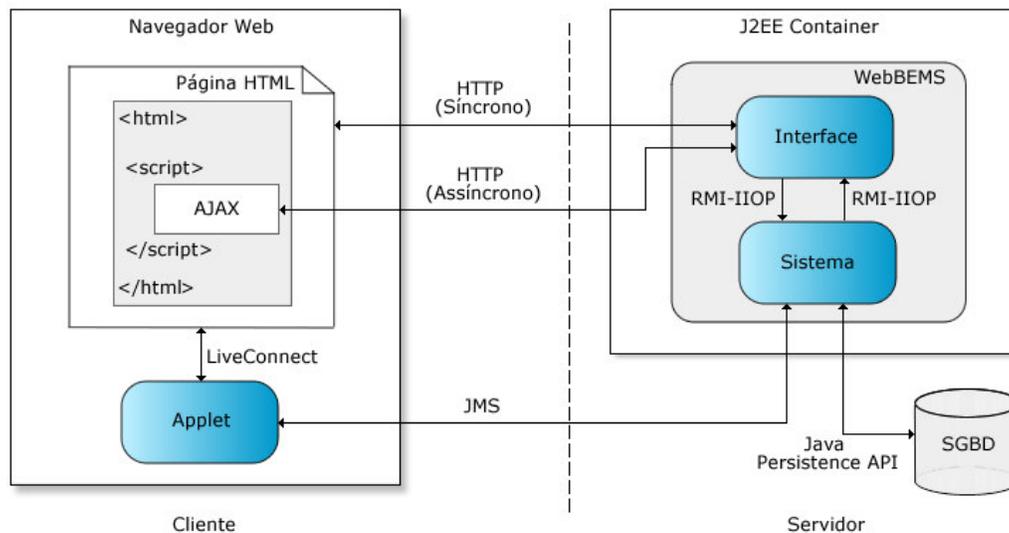
A natureza dinâmica das reuniões exige que os participantes sejam notificados sobre as ações dos demais participantes de uma reunião em tempo real (suporte à percepção). Para atender a este importante requisito a interface foi projetada de modo a tornar o WebBEMS capaz de operar como uma aplicação rica para Internet (*Rich Internet Application*). Neste sentido, a interface vai além de uma interface HTML padrão, limitada a executar requisições e processar respostas. Ela possui elementos ativos que a tornam mais rica e complexa.

O primeiro elemento presente na interface é um motor para execução de *Ashynchronous JavaScript And XML (AJAX)*, que em linhas gerais é uma tecnologia que permite a execução de requisições HTTP assíncronas controladas programaticamente. Este recurso torna a interface capaz de enviar e receber informações ao sistema sem a necessidade de atualizar todo o conteúdo das páginas HTML carregadas pelo navegador Web, reduzindo o tempo de espera e enriquecendo a interação dos usuários.

O outro elemento aplicado na interface é um *Java Applet*, responsável por orquestrar o envio e recebimento de mensagens através da tecnologia *Java Message Service (JMS)*. Este *Applet* não emprega nenhum tipo de recurso visual baseado em *frameworks* de janelamento *Java*, todas as interações entre o usuário e o *Applet* são feitas por meio de *links* e controles presentes na própria página HTML, graças ao uso da tecnologia *LiveConnect* suportada pelos principais navegadores.

Com estes dois elementos combinados, a interface central do WebBEMS é formada por uma única página HTML que embarca dentro de sua estrutura o *Applet*, formato conhecido como *Single Page Application (SPA)*. Uma SPA é uma aplicação web executada completamente dentro de um navegador web.

Uma vez acessada a página, o que demanda um processo de autenticação, todos os conteúdos referentes aos convites, participantes, enquetes e mensagens de bate-papo são aplicados dinamicamente à página usando o motor *AJAX* sem a necessidade de mudar para outra página ou atualizar a página atual. A Figura 4 apresenta a representação completa dos principais elementos que compõe a aplicação e a infraestrutura de comunicação utilizada pelos mesmos.



**Figura 4. Arquitetura de comunicação do WebBEMS**

## 4. Implementação do WebBEMS

### 4.1. Autenticação e autorização

O método de autenticação e autorização implementado no sistema baseia-se na propagação de credenciais do cliente para o servidor a cada requisição. O método de

autenticação e autorização baseada em papéis não foi utilizado porque no WebBEMS o papel de um usuário pode variar indefinidamente entre administrador e participante de uma reunião, o que inviabiliza a atribuição de papéis durante o processo de autenticação.

A autenticação é dividida em duas fases complementares:

- na primeira fase é feita a aquisição das credenciais do usuário através de um formulário de *login*. Estas credenciais são verificadas através de um método de autenticação do sistema, independente do *Java Authentication and Authorization Service (JAAS)*, e, caso sejam válidas, estas são registradas na sessão do usuário que é redirecionado para a área restrita da interface identificada por um URL diferenciado da área não restrita;
- dentro da área restrita, um filtro intercepta todas as requisições a fim de recuperar as credenciais do usuário presentes na sua respectiva sessão, para efetuar o processo de autenticação usando JAAS, que se encarrega de propagar as credenciais, já validadas pelo sistema.

A autorização usa as credenciais propagadas pela interface para confrontar as chamadas de um solicitante e verificar se o mesmo pode executar aquela operação. Por exemplo, um convite para uma reunião só pode ser enviado se o solicitante for o administrador da reunião em questão, do mesmo modo que um convite só pode ser confirmado pelo seu destinatário.

#### **4.2. Transferência de mensagens**

A transferência de mensagens (quer sejam mensagens de bate-papo ou convites) é outra parte importante do sistema e necessita de confiabilidade. No entanto a propagação de credenciais fornecidas pelo JAAS não funciona no sistema de mensagens J2EE, isto é, não é possível saber qual usuário enviou uma mensagem para o sistema e verificar se este possui autorização para executar a operação solicitada.

Uma das formas encontradas para contornar esta questão foi enviar a identificação do usuário dentro das mensagens. Para isto foi criada uma classe usada exclusivamente para que os usuários possam enviar mensagens para o sistema. Desta forma é possível encapsular os dados da mensagem juntamente com a identificação do usuário que a enviou.

Esta estratégia resolve a troca de mensagens que partem dos clientes para o WebBEMS. Contudo a troca de mensagens no sentido inverso deve ser tratada de outra forma. Além de identificar o destinatário da mensagem é preciso que uma mensagem seja consumida apenas pelo seu destinatário e ninguém mais. Para atender a este requisito cada cliente cria uma fila de mensagens temporária. Esta fila temporária é provida pelo JMS e funciona da mesma forma que uma fila convencional, a diferença é que apenas o criador a fila pode ler as mensagens enviadas para a mesma.

Deste modo sempre que o *Applet* é iniciado ele cria uma fila temporária e registra a mesma no WebBEMS para que este saiba qual a fila que cada usuário está usando para receber suas mensagens vindas do sistema. Antes de enviar uma mensagem para um usuário o sistema recupera a fila que está sendo usada pelo usuário e envia a mensagem para a mesma.

### 4.3. Interface web

A interface possui suporte a múltiplos idiomas e é totalmente baseada em HTML. Ela é composta por quadros projetados para se assemelharem com pequenas janelas, através das quais os usuários invocam os serviços do WebBEMS. Ao todo são seis quadros, dispostos simultaneamente na tela, representando as reuniões, os participantes, as enquetes, a sala de bate-papo, os convites recebidos pelo usuário e confirmações de convites enviados pelo usuário. A interface permite que o usuário maximize e minimize o conteúdo de cada quadro, de acordo com sua necessidade. Todos os formulários para cadastro, edição e mensagens são embutidos dentro dos quadros dando-lhes um aspecto de interfaces independentes semelhantes a *Portlets*.

A Figura 5 apresenta uma janela contendo a interface com todos os quadros maximizados e com os respectivos conteúdos a disposição do usuário, que permitem que este tenha percepção das interações dos demais usuários. Caso o usuário deseje se concentrar em apenas determinados quadros ele pode minimizar os quadros que não são de interesse no momento. Por exemplo, durante o cadastro de uma nova votação é possível manter apenas alguns quadros maximizados evitando com isso possíveis distrações. A Figura 6 ilustra esta situação. Nesta figura, os quadros de convites e confirmações estão minimizados.

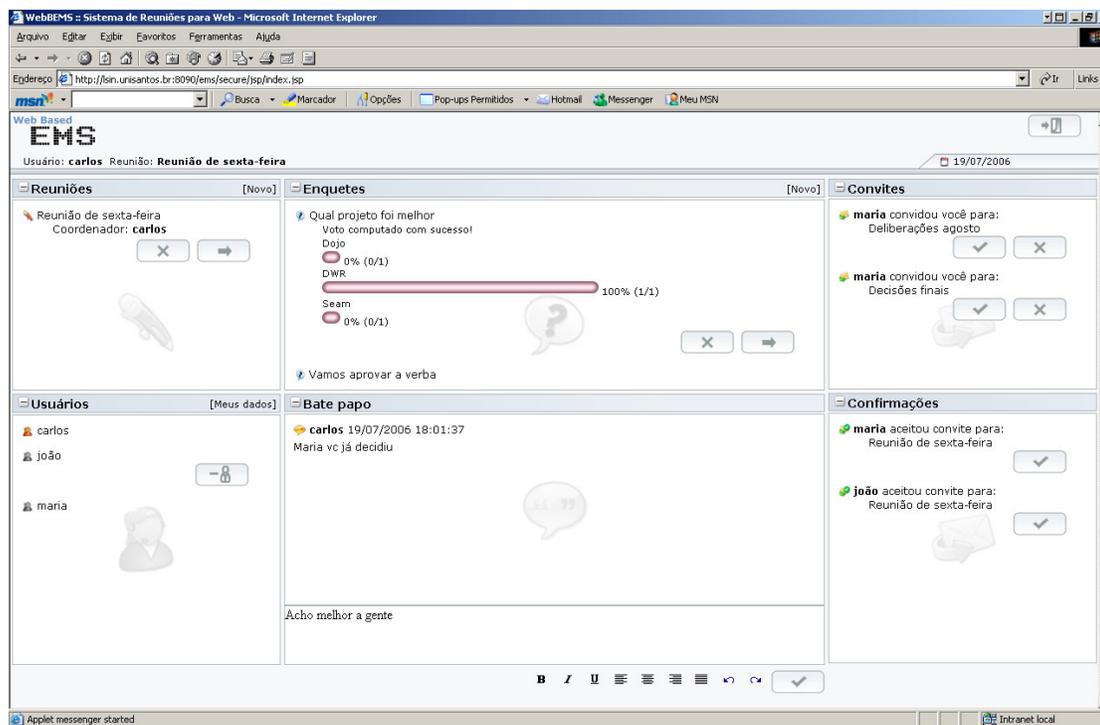


Figura 5. Interface do WebBEMS

Para acessar o conteúdo dinâmico dos quadros e submeter os dados para o servidor a interface utiliza a tecnologia *AJAX*. O motor usado na interface, chamado *Clean AJAX Engine*, simplifica significativamente o processo de construção dos objetos *JavaScript* necessários para a utilização desta tecnologia em diferentes navegadores.

Alguns quadros utilizam o serviço de mensagens *JMS* para acessar e enviar dados para o servidor. Estes quadros utilizam a tecnologia *LiveConnect* para receber e enviar dados ao *Applet* embarcado na página que possui acesso ao serviço *JMS*. O *Applet* possui assinatura via certificado digital para permitir a habilitação de recursos normalmente inacessíveis para *Applets* executados dentro de navegadores.

As informações acessadas através do motor *AJAX* trafegam entre o servidor e o cliente já no formato *HTML*, sem a necessidade de nenhuma transformação no momento da exibição das informações no cliente. Já as informações acessadas pelo *Applet* trafegam entre o servidor e o cliente na forma de objetos *Java*. A exibição destas informações utiliza *templates HTML* disponibilizados com o *Applet*, onde as informações são inseridas para que possam ser exibidas adequadamente.

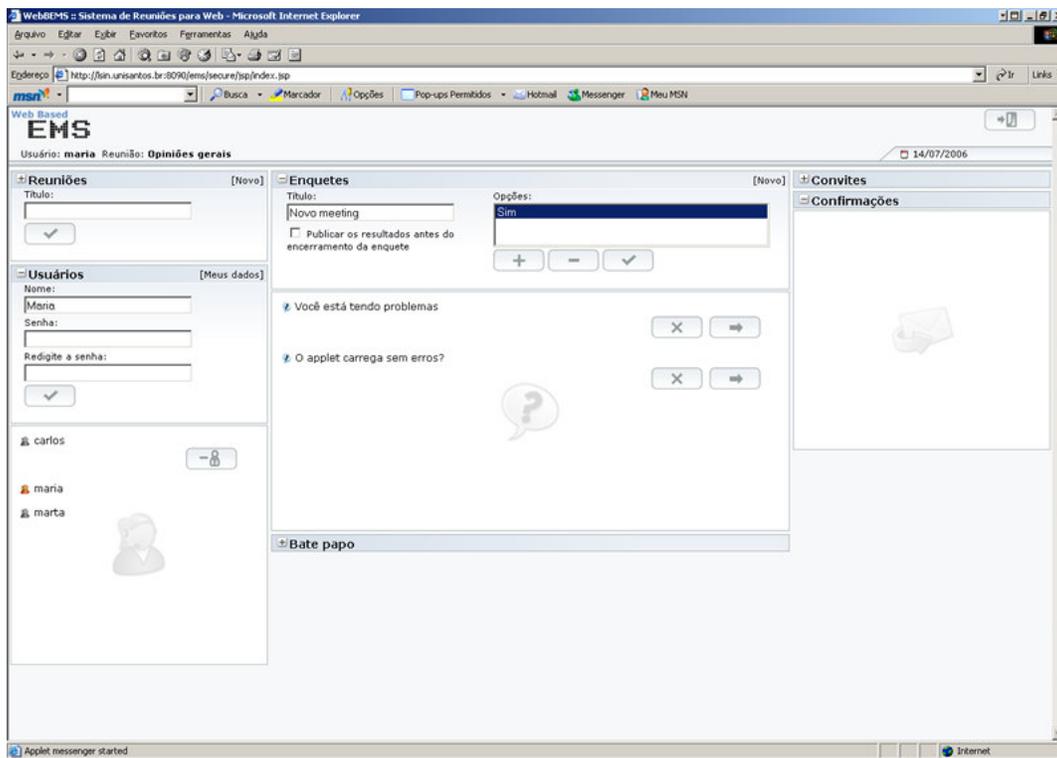


Figura 6. Interface para o cadastro de enquete

#### 4.4. Testes

Os testes de unidade concentraram-se nos componentes EJB e foram executados utilizando a prática de testes *in-container*, a partir do *framework Cactus*, na qual os testes de unidade são executados dentro do ambiente provido pelo EJB *container*. Os testes de aceitação foram feitos a partir da simulação de reuniões observando o comportamento e performance da aplicação como um todo, isto é, observando os recursos providos tanto pelo lado servidor quanto pelo lado cliente.

## 5. Trabalhos Relacionados

Sistemas de reunião eletrônica são sistemas colaborativos concebidos para prover suporte eletrônico a diferentes tipos de reuniões, desde reuniões síncronas face-a-face até reuniões assíncronas distribuídas [Nunamaker *et al.* 1991]. Em geral estes sistemas têm por objetivo aumentar a produtividade e satisfação dos participantes envolvidos nestas atividades. Tais sistemas podem, por exemplo, dar suporte ao aprendizado colaborativo no contexto educacional ou dar suporte ao trabalho cooperativo e a tomada de decisão de maneira geral.

No contexto da educação, enfatiza-se o valor dos sistemas de suporte a grupos em geral, definindo-o como um sistema computacional que permite que um grupo de pessoas tanto possa entrar com dados como também compartilhá-los [Weatherall 2000]. Este tipo de sistema tem sido usado para melhorar a efetividade de reuniões face-a-face, onde a discussão é suplementada pela possibilidade dos participantes colocarem fatos, opiniões, questões, votos, etc. Nesta situação, geralmente cada participante da reunião tem seu próprio computador. Já no cenário de reunião eletrônica, um sistema de suporte a colaboração permite que os participantes possam participar de diferentes atividades a partir do trabalho, de casa, etc., o que é particularmente relevante no contexto de um treinamento.

Em [de Farias *et al.* 2005, de Farias *et al.* 2006] mostrou-se a relevância de um modelo de arquitetura baseada em componentes para os sistemas e plataformas de aprendizado eletrônico. A aplicação de um modelo de componentes projetado especificamente para atender sistemas colaborativos foi ilustrada através da implementação de um sistema de votação eletrônica baseado na Web, denominado Electronic Voting System (EVS). O EVS basicamente permite que os seus usuários criem uma enquete e/ou votem nela. No aprendizado eletrônico o sistema permite que os estudantes incluam anonimamente idéias e preferências através da criação de uma enquete, votem secretamente e verifiquem a existência de consenso ou não no grupo sobre uma determinada questão, apontando os pontos que devem ser discutidos ou debatidos [Yankelovich *et al.* 2004].

O WebBEMS é um sistema de reunião eletrônica que foi desenvolvido com o objetivo de contemplar os principais recursos que permitem dar suporte e ampliar a colaboração entre os usuários. O sistema pode ser utilizado em diferentes contextos mas especialmente no aprendizado eletrônico permite a utilização de abordagens pedagógicas onde a discussão e a busca por uma posição de consenso ou de maioria sobre uma determinada questão é necessária. O WebBEMS combina as funcionalidades de um sistema de votação eletrônica (i.e., o EVS) com as funcionalidades básicas de uma ferramenta de bate-papo para dar suporte a este tipo de atividade. Como apontado por [Kennedy and Cutts 2005], a adição do componente de bate-papo dá a oportunidade aos participantes de uma atividade colaborativa, especialmente no contexto educacional, de discutirem as questões e suas respostas uns com os outros, associando a votação com uma discussão com os seus pares.

## 6. Conclusão

Este artigo apresentou o projeto e o desenvolvimento do WebBEMS, um sistema baseado em componentes para reuniões eletrônicas pela Internet. Toda a especificação e

desenvolvimento do WebBEMS foi baseado em um ciclo de desenvolvimento incremental e iterativo, que conferiu grande dinâmica e agilidade ao projeto.

O projeto do WebBEMS foi norteado por um modelo arquitetônico que têm sido proposto para o desenvolvimento de sistemas colaborativos baseados em componentes. A adoção deste modelo facilitou a identificação e estruturação dos componentes ao longo do processo de especificação e implementação do sistema. Adicionalmente, utilizamos diversos padrões de projeto de modo a oferecer uma grande flexibilidade de manutenção e reengenharia ao sistema. A interface com usuário do WebBEMS possui um design atual voltado a aplicações Web 2.0 e que promove uma maior interatividade entre os usuários se comparada com as interfaces Web tradicionais.

No que se refere ao reuso dos componentes em comum entre o EVS [de Farias *et al.* 2005, de Farias *et al.* 2006] e o WebBEMS, que são os componentes *UserManager* e *PollManager*, considera-se que o nível de reutilização foi baixo. Isto se deve ao fato de que ocorreram alterações significativas nestes componentes com o acréscimo de novas funcionalidades. No componente *UserManager* o EJB responsável por controlar os usuários logados no sistema havia sido implementado no EVS com base no padrão *Singleton*. Contudo o padrão *Singleton* é um padrão desaconselhado para a os componentes EJB, e isto motivou a reconstrução de parte do *UserManager*. No componente *PollManager*, a inclusão de novos requisitos funcionais ampliou significativamente o tamanho do componente, o que motivou o seu desdobramento em 3 componentes no WebBEMS (*PollManager*, *SessionManager*, e *ParticipantManager*).

Em princípio, pretende-se utilizar o WebBEMS no aprendizado eletrônico, com suporte a interações presenciais e à distância. Adicionalmente, planeja-se ampliar as funcionalidades do sistema através da adição de serviços de calendário em grupo e filtro de mensagens.

## **Agradecimentos**

Trabalho realizado com o auxílio da FAPESP sob o projeto número 2003/08279-2.

## **Referências**

- Bass, L., Clements, P. and Kazman, R. (1997). *Software Architecture in Practice*. Addison-Wesley.
- Brusilovsky, P. (2004). KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. *Proceedings of the 13th international World Wide Web Conference (WWW'04)*, pp. 104-113.
- de Farias, C. R.G. (2002). *Architectural Design of Groupware Systems: a Component-Based Approach*. PhD Thesis, University of Twente, Enschede, the Netherlands.
- de Farias, C.R.G., Gonçalves, C.E., Rosatelli, M.C., Ferreira Pires, L. and van Sinderen, M. (2005). An Architectural Model for Component Groupware. *Proceedings of the 11th Int. Workshop on Groupware (CRIWG'05)*, Lecture Notes in Computer Science, 3706, Springer-Verlag, pp. 105-120.
- de Farias, C.R.G., Rosatelli, M.C., Gonçalves, C.E. (2006). Applying a Component-Based Architectural Model in the Development of e-Learning Systems. *Proceedings*

- of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT'06), IEEE Computer Society Press, pp. 247-251.
- Falkner, J., Jones, K. (2003). *Servlets and JavaServer Pages: The J2EE Technology Web Tier*. Addison-Wesley.
- Fuks, H., Raposo, A. B., Gerosa, M. A. and Lucena, C. J. P. (2005). Applying the 3C Model to Groupware Development. In *International Journal of Cooperative Information Systems (IJCIS)*, 14(2-3), pp. 299-328
- Hightower, R. (2004). *Jakarta Struts Live*. SourceBeat.
- Jeffries, R., Hendrickson, C., Anderson, A. (2000). *Extreme Programming Installed*. Addison-Wesley.
- Kennedy, G. E. and Cutts, Q. I. (2005). The Association Between Students' Use of an Electronic Voting System and Their Learning Outcomes. In *Journal of Computer Assisted Learning*, 21(4), Pp. 260-268.
- Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R. and George, J.F. (1991). Electronic meeting systems to support group work. *Communications of the ACM*, 34 (7), pp. 40-61.
- OMG (2007a): *Unified Modeling Language: Infrastructure*. OMG Specification, version 2.1.1, Object Management Group.
- OMG (2007b): *Unified Modeling Language: Superstructure*. OMG Specification, version 2.1.1, Object Management Group.
- Pukkhem, N., Vatanawood, W. (2005). Instructional Design Using Component-Based Development and Learning Object Classification. *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, p. 492-494.
- Sriganesh, R. P., Brose, G. and Silverman, M. (2006). *Mastering Enterprise JavaBeans 3.0*. Wiley Publishing Inc.
- ter Hofte, G. H. (1998): Working Apart Together: Foundations for Component Groupware. PhD Thesis, Telematics Institute, the Netherlands.
- van Vliet, H. (2000). *Software Engineering: Principles and Practice*. John Wiley & Sons, USA.
- Weatherall, A. (2000). Improving and Focusing a Training Course using GroupSystems Electronic Meetings. Proceedings of the 33rd Hawaii International Conference on System Sciences, IEEE Computer Society Press, pp. 292- 300.
- Yankelovich, N., Walker, W., Roberts, P., Wessler, M., Kaplan, J., and Provino J. (2004). Meeting central: Making Distributed Meetings More Effective. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW'04)*, ACM Press, pp. 419-428.

# Uma comparação técnica das abordagens existentes para implementação de requisitos colaborativos em aplicações não colaborativas

Mauro C. Pichiliani<sup>1</sup>, Celso M. Hirata<sup>1</sup>

<sup>1</sup>Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)  
Caixa Postal 12.228-900 - São José dos Campos - SP

{pichilia, hirata}@ita.br

**Abstract.** *Many approaches can be used to facilitate the implementation of collaborative requirements on non collaborative applications. Each approach uses different design techniques and has specific objectives and pre-conditions. During the implementation of collaborative requirements is not always easy to decide which approach is the most recommended i.e. which criteria should be used to compare the approaches. Based on the literature work, this paper presents a set of technical criteria to technically compare the existing approaches to implement collaborative requirements on non collaborative applications. With these criteria, the developers can choose the most suitable approach to implement collaborative requirements on non collaborative applications.*

**Resumo.** *Existem diversas técnicas para facilitar a implementação de requisitos colaborativos em aplicações não colaborativas. Cada abordagem usa técnicas de projeto diferentes e possui pré-requisitos e objetivos específicos. Na implementação dos requisitos colaborativos, nem sempre é imediato decidir qual é a abordagem mais recomendada i.e. quais devem ser os critérios comparativos. A partir do estudo bibliográfico, este trabalho apresenta um conjunto de critérios para comparar tecnicamente as abordagens existentes. Deste modo, espera-se que a comparação técnica e os critérios apresentados neste trabalho auxiliem os desenvolvedores na implementação de requisitos colaborativos em aplicações não colaborativas.*

## 1. Introdução

A implementação de requisitos colaborativos em aplicações não colaborativas possui o potencial de aumentar significativamente a disponibilidade de aplicações colaborativas e melhorar a usabilidade dessas aplicações [Xia et al. 2004]. Durante a última década, vários esforços na área de CSCW (*Computer Supported Collaborative Work*) contribuíram para a redução da complexidade necessária para estender e adaptar aplicações não colaborativas, com o objetivo de apoiar a colaboração.

Atualmente, existe uma variedade de abordagens de extensão e adaptação de aplicações não colaborativas, contudo, não se dispõe de critérios de comparação das abordagens disponíveis para implementar requisitos colaborativos em aplicações não colaborativas. Isto se deve, em parte, à heterogeneidade das abordagens, que variam desde a utilização de aplicações externas para emular eventos até a substituição de componentes da aplicação.

Os principais requisitos colaborativos implementados em aplicações colaborativas são: (i) comunicação, utilizada para trocar idéias, discutir, aprender, negociar ou para tomar decisões; (ii) coordenação, que organiza o grupo para evitar que esforços de comunicação e cooperação sejam perdidos e que as tarefas sejam realizadas na ordem e no tempo correto; (iii) percepção (*awareness*), que faz com que um participante fisicamente separado esteja ciente da presença e das ações dos demais participantes do grupo; e (iv) compartilhamento de informações, encarregado de prevenir esforços repetitivos e assegurar que todos os participantes do grupo estejam utilizando a mesma informação.

Critérios técnicos de comparação de abordagens podem auxiliar os desenvolvedores na escolha da abordagem a ser utilizada, além de fornecer uma contextualização do uso de cada uma das abordagens. Deste modo, o desenvolvedor pode compreender melhor qual tipo de abordagem é mais recomendada na implementação dos requisitos colaborativos desejados, de acordo com seu contexto.

O objetivo deste artigo é apresentar um conjunto de critérios para as abordagens de implementação de requisitos colaborativos e uma comparação técnica das abordagens utilizando os critérios de comparação. A partir da comparação de abordagens apresentada neste trabalho, os desenvolvedores podem analisar a conveniência de uma determinada abordagem. Tendo em vista que a implementação de requisitos colaborativos apresenta vários desafios, possuir uma comparação técnica de abordagens facilita o processo de desenvolvimento.

Este artigo está dividido em quatro seções. A Seção 2 descreve as abordagens utilizadas para a implementação de requisitos colaborativos em aplicações existentes. Na Seção 3 apresentam-se os critérios comparativos e a comparação técnica propriamente dita. Por fim, na Seção 4 são apresentadas as conclusões e alguns comentários finais.

## **2. Abordagens Existentes**

Esta seção descreve as abordagens existentes na literatura para construir aplicações colaborativas e para transformar aplicações não colaborativas existentes em aplicações colaborativas. As abordagens descritas nesta seção incluem o uso de *toolkits*, sistemas de colaboração transparente, substituição dos componentes da aplicação por componentes colaborativos, adaptação transparente e mapeamento de componentes.

A estratégia tradicional para a modificação de uma aplicação, isto é, a modificação da aplicação de forma *ad hoc*, não será abordada nesta comparação, pois esta abordagem proporciona pouca ou nenhuma técnica sistemática de desenvolvimento que possa ser seguida para transformar aplicações não colaborativas existentes em aplicações colaborativas.

### **2.1 Toolkits**

Numa tentativa de reduzir a complexidade de implementação de sistemas *groupware*, vários pesquisadores têm explorado o uso de *toolkits*. Segundo Ellis et al. [Ellis et al. 1991], sistemas *groupware* são sistemas baseados em computadores que apóiam grupos de pessoas engajadas em uma tarefa ou objetivo comum e que provém uma interface para um ambiente compartilhado.

Um *toolkit* é definido como um conjunto de componentes predefinidos e reutilizáveis, com a finalidade de oferecer ferramentas e infra-estrutura suficientes para permitir que o programador desenvolva sistemas *groupware* de alta qualidade com razoável esforço [Greenberg & Roseman 1998].

Os *toolkits* para a construção de sistemas *groupware* facilitam a criação de novas aplicações colaborativas por meio de componentes de grupo e controles de percepção construídos para serem reutilizáveis, permitindo a criação de poderosas aplicações colaborativas. Da mesma forma que casas pré-fabricadas podem ser construídas a partir da montagem de módulos pré-fabricados, o uso de *toolkits* na construção de software colaborativo pode tornar a tarefa do desenvolvedor mais simples e, muitas vezes, mais rápida, uma vez que o trabalho principal consiste apenas na montagem e configuração dos componentes em vez da criação deles a partir do zero.

Devido a estas características, pode-se dizer que o uso de *toolkits* para a construção de sistemas *groupware* segue, de uma maneira geral, os princípios envolvidos no Desenvolvimento Baseado em Componentes.

O Desenvolvimento Baseado em Componentes (DBC) têm como objetivo prover o reuso dos componentes no nível de implementação e aumentar a interoperabilidade entre as partes do software. Os princípios envolvidos no Desenvolvimento Baseado em Componentes demandam características específicas dos componentes, como a capacidade de prover funcionalidades específicas, a baixa dependência entre componentes e a suscetibilidade a modificações.

*Toolkits* amplamente utilizados, como o GroupKit [Gutwin & Greenberg 2002], o COAST (*COoperative Application Systems Toolkit*) [Schuckmann et al. 1996] e o MAUI (*Multi-User Awareness UI toolkit*) [Hill & Gutwin 2004], fornecem *frameworks*, ambientes, bibliotecas de funções, componentes e controles para tornar mais rápida e fácil a construção de novas aplicações colaborativas.

O reuso dos componentes de *toolkits* pode ajudar o desenvolvimento de novos softwares colaborativos, entretanto, os componentes fornecidos pelo *toolkit* não podem auxiliar a adaptação de uma aplicação não colaborativa existente. Em geral isso acontece devido à necessidade de implementação de detalhes específicos do *toolkit*, que nem sempre são compatíveis com a estrutura da maioria das aplicações existentes.

Um exemplo desta incompatibilidade é a diferença entre a estrutura de uma aplicação existente a estrutura do *toolkit*. Além disso, detalhes técnicos como a linguagem de programação e a arquitetura utilizada na construção dos componentes do *toolkit* podem tornar o reuso dos componentes inviável em certas situações. Mesmo nos casos onde os detalhes técnicos não inviabilizam a utilização do *toolkit*, a quantidade de esforço necessário para a reutilização dos componentes pode não compensar.

Apesar dos *toolkits* não serem muito úteis para ajudar a extensão de um software existente, eles podem fornecer inspiração e idéias necessárias para implementar aspectos essenciais de comunicação, colaboração e cooperação em softwares existentes.

## **2.2 Sistemas de Colaboração Transparente**

Os Sistemas de Colaboração Transparente (SCT) fornecem um meio de compartilhar o uso de aplicações existentes apenas de forma síncrona, onde as interações dos participantes são simultâneas ou separadas por pequenos períodos de tempo. Estes

sistemas permitem apenas a forma síncrona de interação entre os participantes, pois eles emulam as entradas e saídas de dados dos usuários da aplicação, fornecendo aos usuários a impressão de que todos utilizam a mesma interface da aplicação, ao mesmo tempo e no mesmo local. De acordo com Begole et al. [Begole et al. 1999], o termo transparente é utilizado por estes sistemas porque o compartilhamento fornecido é transparente ou desconhecido para a aplicação e seus desenvolvedores. O compartilhamento, neste contexto, envolve as aplicações legadas com interação monousuária.

Estes sistemas utilizam o conceito de caixa preta de colaboração transparente, onde alguma aplicação externa é responsável pelas comunicações e notificações necessárias para apoiar aspectos básicos da colaboração entre os usuários remotos. Os mecanismos por trás desses sistemas incluem uma arquitetura centralizada, a captação e entrega dos dados de entrada dos usuários para múltiplos locais e a transmissão da interface gráfica da aplicação para todos os participantes. Exemplos de SCT comerciais incluem o Microsoft Netmeeting [Microsoft 2007], o XTV [Abdel-Wahab & Feit 1991] e o SharedX [Garnkel et al. 1994].

Colaboração Transparente Inteligente, ou ICT (*Intelligent Collaboration Transparency*) [Li & Li 2002], apresenta uma evolução dos SCT, pois eles utilizam uma infra-estrutura compartilhada entreposta entre aplicações heterogêneas e o sistema operacional para compartilhar os ambientes gráficos de cada local.

Devido à heterogeneidade das aplicações envolvidas, a simples reprodução dos eventos de uma aplicação em outras aplicações diferentes não faz sentido. Uma operação semântica de uma aplicação geralmente é implementada de forma diferente em outra aplicação, requerendo um mecanismo que, além de compreender as operações dos usuários, deve traduzir essas operações em seqüência de operações equivalentes a outras aplicações, antes que elas sejam reproduzidas. Deste modo, o mecanismo deve ser inteligente o suficiente para traduzir as operações de uma aplicação em uma seqüência de operações compatíveis em outras aplicações, por meio do conhecimento semântico das ações realizadas pelos usuários.

Com o uso deste tipo de sistemas, os usuários podem colaborar em uma tarefa comum, através de suas aplicações favoritas. Contudo, um módulo de captura/reprodução de eventos específico para cada plataforma deve ser utilizado para tratar da comunicação e do controle de concorrência, propiciando a interoperabilidade entre as aplicações.

Uma segunda geração de aplicações baseadas na abordagem do ICT foi elaborada por Lu et al. [Lu et al 2004], denominada ICT2. A principal diferença entre a abordagem ICT clássica e a ICT2 é que a última não intercepta e interpreta os eventos da aplicação. Em vez disso, a ICT2 utiliza uma versão adaptada de um algoritmo que gera as seqüências de edição entre os diferentes estados dos documentos locais dos usuários. Para organizar as ações concorrentes dos usuários, uma versão modificada do algoritmo de transformação operacional é utilizada.

Mangan [Mangan 2006] propõe uma variação da técnica de colaboração transparente. Esta proposta chama-se Arquitetura de Colaboração Transparente (ACT) e amplia as arquiteturas transparentes com o acréscimo do conceito de eventos semânticos. Neste contexto, os eventos descrevem o significado das operações realizadas pelo usuário da aplicação e são gerados a partir de interpretações dos eventos

de entrada e dos eventos de aplicação das arquiteturas transparentes. Com estes eventos, é possível alimentar os modelos de percepção que oferecem informações úteis para mecanismos: de apoio à colaboração assíncrona; de descoberta de perfil de usuário; e de histórico das experiências dos usuários.

Apesar de apresentar uma alternativa rápida para compartilhar aplicações sem modificações no código fonte, os SCTs da primeira geração, a ACT e as abordagens ICT e ITC2 receberam muitas críticas no sentido que elas não suportam a colaboração adequadamente. O motivo principal dessas críticas envolve o uso ineficiente dos recursos de rede e a imposição de um estilo de colaboração inflexível e altamente acoplado que não suporta algumas funcionalidades comuns aos sistemas *groupware* como, por exemplo, controle de concorrência, interface WYSIWIS (*What You See Is What I See*) relaxada, percepção em grupo e tarefas colaborativas delegadas [Begole et al. 1999]. Uma interface WYSIWIS relaxada, em particular, propicia aos usuários o controle sobre aspectos de suas interface individuais permitindo que eles trabalhem de forma mais natural, sendo considerada uma funcionalidade determinando para a ação de um *groupware*.

### **2.3 Adaptação Transparente**

A abordagem chamada Adaptação Transparente foi proposta para ajudar a implementar a colaboração em aplicações comerciais, quando os desenvolvedores não têm acesso ao código fonte da aplicação. Esta abordagem é baseada no compartilhamento da aplicação e no uso de uma Interface de Programa de Aplicação (API) para interceptar as interações locais do usuário, convertê-las em operações abstratas, manipular estas operações por técnicas colaborativas e reproduzir as operações modificadas, por meio da API, nos locais remotos de colaboração [Xia et al. 2004].

O termo transparente é utilizado porque esta abordagem não requer nenhuma mudança no código fonte da aplicação. Contudo, esta abordagem requer que o fabricante da aplicação forneça uma API capaz de gerenciar os dados da aplicação e implementar os mecanismos de colaboração.

Diferentemente dos ambientes de compartilhamento de aplicações, os quais não requerem nenhum tratamento específico, a abordagem de Adaptação Transparente requer uma nova camada de software para implementar a arquitetura replicada que fornecerá o compartilhamento de eventos e dados da aplicação. De acordo com Xia et al. [Xia et al. 2004], a combinação da arquitetura replicada e da abordagem de Adaptação Transparente permite vários benefícios como, por exemplo, uma grande quantidade de interações entre os usuários, um controle de concorrência eficiente e uma melhor interface WYSIWIS relaxada.

Desta maneira, o código fonte da aplicação não é necessário. Contudo, uma API completa e que forneça o acesso aos eventos e dados gerados pela aplicação deve ser fornecida. Mesmo com uma API completa, alguns métodos específicos de controle de concorrência e dispositivos de presença são impossíveis de implementar na abordagem de Adaptação Transparente, devido à necessidade da mudança no código fonte da aplicação para implementar alguns requisitos colaborativos.

## 2.4 Substituição de Componentes

A abordagem de substituição de alguns componentes de uma aplicação por componentes colaborativos foi introduzida pelo Flexible JAMM (*Java Applets Made Multi-user*) [Begole et al. 1999], sendo conhecida também como Sistema de Colaboração Transparente Flexível. Na comparação apresentada neste trabalho, esta abordagem é denominada Substituição de Componentes, pois esta descrição identifica rapidamente a técnica utilizada na implementação da abordagem.

Esta abordagem permite que os usuários trabalhem juntos de forma colaborativa ou independente, por meio da substituição dinâmica de certos objetos da interface do usuário por componentes de interfaces colaborativas [Begole et al. 1999]. Para utilizar esses componentes colaborativos, a aplicação que recebe os componentes deve possuir certas características como, por exemplo, a capacidade de migração de processos, recursos para a substituição de componentes em tempo de execução e a habilidade para interceptar e reproduzir os eventos gerados pelo usuário.

Os componentes da interface gráfica de usuário que o Flexible JAMM fornece incluem mecanismos de percepção como barras de rolagem multiusuário, visão radar e *telepointers*. Estes componentes propiciam, respectivamente, nomes de usuários em diferentes barras de rolagem, visão em miniatura da área de trabalho comum e ponteiros virtuais representando usuários remotos.

A principal limitação desta abordagem reside no fato que os componentes oferecidos pelo Flexible JAMM suportam apenas o requisito colaborativo percepção, não apresentando componentes para os requisitos colaborativos comunicação, coordenação e compartilhamento de informações. Além disso, somente um pequeno grupo de aplicações existentes possui todos os requisitos necessários para o uso dos componentes do Flexible JAMM, ou seja, apenas as aplicações que atendam aos requisitos de migração de processos, substituição de componentes em tempo de execução e a habilidade para interceptar e reproduzir os eventos podem utilizar os componentes oferecidos pelo Flexible JAMM. Estas características limitam o uso do Flexible JAMM a poucas aplicações, evitando que esta abordagem possa ser utilizada de forma sistemática em softwares de diferentes domínios de conhecimento.

## 2.5 Mapeamento de Componentes

O Mapeamento de Componentes, proposto por Pichiliani & Hirata [Pichiliani & Hirata 2006], sugere um mapeamento dos principais componentes de aplicações não colaborativas, que devem estar baseadas no estilo arquitetural MVC (*Model-View-Controller*), para componentes de uma aplicação colaborativa com o objetivo de permitir a colaboração síncrona entre participantes. Usando o mapeamento, aplicações existentes podem ser estendidas para apoiar a colaboração síncrona durante a elaboração de tarefas compartilhadas.

De acordo com os autores, o Mapeamento de Componentes permite um tratamento uniforme de um conjunto de requisitos colaborativos mínimos, em um nível alto de abstração, com o objetivo de fornecer um ponto de partida para os desenvolvedores de aplicações, sob o ponto de vista conceitual.

Esta abordagem requer que a aplicação não colaborativa, donde é feito o Mapeamento de Componentes, estruture seus componentes de acordo com o estilo

arquitetural MVC, que sugere a separação dos componentes que manipulam dados da aplicação (o modelo), dos componentes de tratamento de dados (o controlador) e também dos componentes responsáveis interface de visualização (a visão).

Além de requerer a organização dos componentes da aplicação não colaborativa no estilo arquitetural MVC, o Mapeamento de Componentes implica na modificação do código fonte da aplicação para alterar o comportamento dos componentes, de acordo com os requisitos de colaboração especificados. A partir do uso do mapeamento, a aplicação não colaborativa passa a suportar requisitos colaborativos, compartilhando os dados através de uma arquitetura híbrida, onde um Servidor de Colaboração é responsável pela a comunicação dos dados manipulados pelos componentes do Modelo.

A principal desvantagem do Mapeamento de Componentes é a necessidade de modificação direta do código fonte da aplicação. Esta necessidade envolve diversos recursos que nem sempre são disponíveis, como o código fonte da aplicação ou um compilador compatível com a linguagem de programação da aplicação. Deste modo, o uso desta abordagem se torna inviável em algumas ocasiões. Contudo, pelo fato de se trabalhar no código fonte, esta abordagem oferece uma grande flexibilidade na implementação dos requisitos colaborativos.

### **3. Comparação Técnica de Abordagens**

Esta seção descreve os critérios técnicos utilizados para comparar as abordagens apresentadas na seção anterior. Em seguida, a comparação entre as abordagens que transformam aplicações não colaborativas existentes em aplicações colaborativas é apresentada.

#### **3.1 Critérios de Comparação**

Uma vez que as abordagens existentes foram apresentadas, é necessário comparar suas características por meio de critérios técnicos. Diversos critérios podem ser escolhidos para a comparação, porém os critérios escolhidos nesta comparação entre abordagens visam auxiliar o desenvolvedor, concentrando-se nas características e nos detalhes envolvidos no uso das abordagens, obtidas por meio dos trabalhos relacionados e pelo estudo experimental das abordagens, conduzido pelos autores deste trabalho. Os critérios de comparação são os seguintes: Código Fonte, Requisito Tecnológico, Propósito, Arquitetura e Flexibilidade.

Um critério a ser considerado antes da adoção de alguma abordagem leva em consideração a disponibilidade do código fonte. Este critério está diretamente relacionado com o modelo de licença que a aplicação segue. Os modelos de licença que as aplicações seguem, para efeitos de comparação de abordagens, podem ser classificados, basicamente, em dois tipos: aplicações abertas, isto é, aplicações que possuem uma licença onde o código fonte é disponibilizado publicamente e que permite a modificação sem nenhum ônus; e aplicações proprietárias, onde a licença não permite que o código fonte seja modificado. Na comparação de abordagens apresentada nesta seção, quatro valores auto-explicativos são especificados para o critério Código Fonte, a saber: ‘Requer o código fonte (da aplicação)’; ‘Não requer o código fonte (da aplicação)’; ‘Requer API do sistema operacional’; e ‘Requer API da aplicação’.

Outro critério importante a ser considerado na comparação das abordagens envolve os requisitos tecnológicos relacionados com a implementação. Estes requisitos

influenciam diretamente o uso da aplicação a partir do momento que a abordagem for implementada. Por exemplo, a abordagem de *toolkits* requer uma linguagem de programação específica para que o *toolkit* possa ser utilizado, o que pode causar problemas de portabilidade e integração decorrentes da necessidade deste requisito. O critério Requisito Tecnológico, utilizado na comparação das abordagens realizada nesta seção, pode assumir os seguintes valores: ‘Nenhum’, ‘Depende da linguagem de programação’, ‘Requer camada de software’, ‘Depende da linguagem do componente’ e ‘Requer aplicação no estilo MVC’.

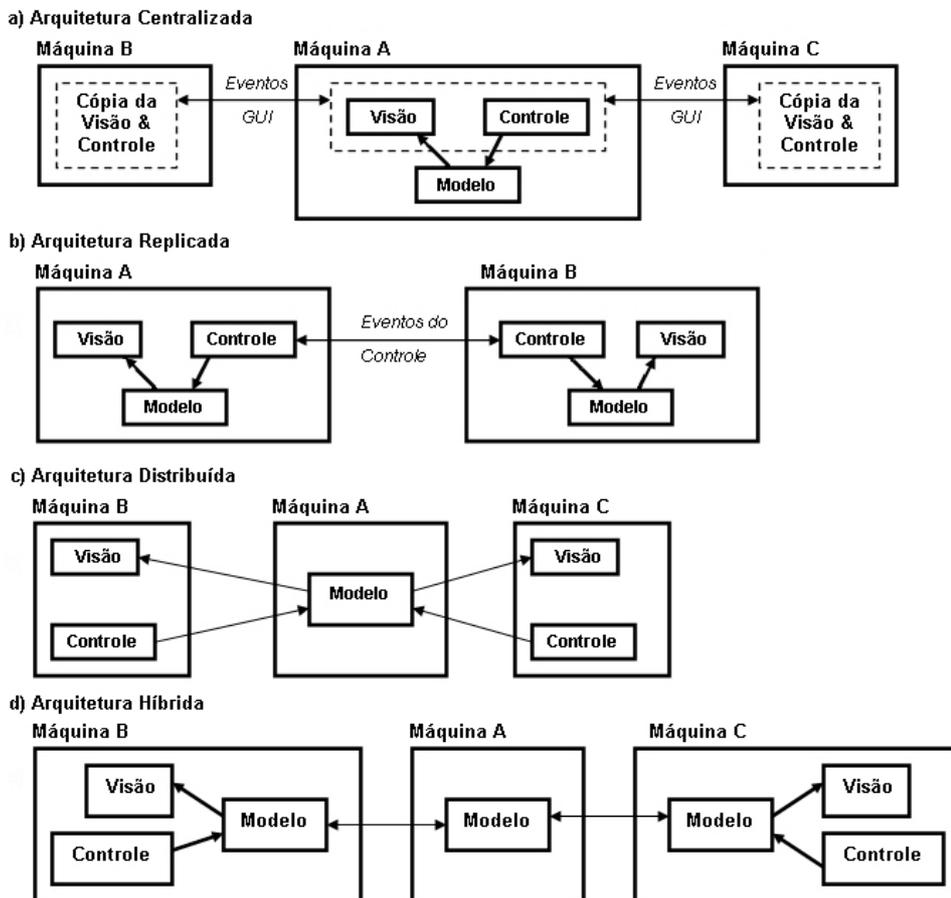
É fundamental compreender o propósito das abordagens antes de considerar sua utilização. Enquanto algumas abordagens demandam a modificação direta do código fonte, outras devem ser consideradas apenas para novas aplicações. No critério Propósito, a abordagem pode possuir o valor “Criar novas aplicações”, para indicar o contexto de construção de novas aplicações por meio da abordagem, ou o valor “Promover colaboração”, para indicar que a abordagem é utilizada para promover a colaboração em uma aplicação já existente. Nota-se que uma abordagem pode conter os dois valores para este critério, como é o caso da abordagem ‘Substituição de Componentes’.

A arquitetura da aplicação colaborativa obtida a partir da utilização de qualquer uma das abordagens apresentadas na Seção 2 é um critério a ser considerado nesta comparação de abordagens. Suthers [Suthers 2001] apresenta algumas maneiras pelas quais as aplicações colaborativas de ensino organizam seus componentes na arquitetura MVC, de acordo com o nível de acoplamento dos componentes, isto é, de acordo com o nível de dependência entre os componentes. Apesar de focar apenas em aplicações colaborativas na área de ensino, as arquiteturas apresentadas por Suthers são genéricas a ponto de poderem ser consideradas para aplicações colaborativas de propósito gerais.

A Figura 1 apresenta os quatro tipos de organização do modelo MVC sugeridos por Suthers: 1) Arquitetura Centralizada; 2) Arquitetura Replicada; 3) Arquitetura Distribuída; e 4) Arquitetura Híbrida. Apesar de não ser exaustiva, as arquiteturas apresentadas por Suthers representam como a maioria dos sistemas, *groupware* ou não, são organizados. Caracterizar a arquitetura da abordagem é importante, pois a arquitetura e os casos de uso orientam o projeto detalhado e a implementação de aplicações no processo de desenvolvimento orientado a objetos.

Deste modo, o critério Arquitetura indica qual das arquiteturas pode ser utilizada a partir da escolha da abordagem, assumindo os valores ‘Centralizada’, ‘Replicada’, ‘Distribuída’ e ‘Híbrida’. De forma análoga ao critério Propósito, uma mesma abordagem pode usar diferentes arquiteturas.

O último critério técnico utilizado na comparação envolve a Flexibilidade das abordagens para implementar requisitos colaborativos. Este critério indica a capacidade que as abordagens detêm para satisfazer requisitos. Este critério pode assumir os valores subjetivos ‘Pouca’, ‘Média’, e ‘Muita’, para indicar que a abordagem apresenta, respectivamente, pouca, média e muita flexibilidade na implementação de requisitos colaborativos. Este critério é superficial, cujo objetivo é apenas indicar de forma geral os diferentes níveis de flexibilidade que cada abordagem proporciona



**Figura 1. Organizações dos componentes do MVC em: a) Arquitetura Centralizada, b) Arquitetura Replicada, c) Arquitetura Distribuída e d) Arquitetura Híbrida. Adaptada de [Suthers 2001].**

As seguintes abordagens, apresentadas na Seção 2, são comparadas neste trabalho: a abordagem *toolkit*, a primeira geração de Sistemas de Colaboração Transparente (SCT), as abordagens ICT, ICT2 e ACT da segunda geração de SCT, a abordagem de Adaptação Transparente, a Substituição de Componentes e o Mapeamento de Componentes. A Tabela 1 apresenta a comparação de abordagens de acordo com os critérios estabelecidos.

### 3.2 Discussão da Comparação Técnica de Abordagens

A abordagem *'Toolkits'* é utilizada para a construção de novas aplicações colaborativas e não permite a promoção de colaboração entre aplicações já existentes. Devido a esta característica, esta abordagem não se aplica no caso de implementação de requisitos colaborativos a partir de uma aplicação não colaborativa. Contudo, para as situações onde uma nova aplicação colaborativa deve ser construída, a abordagem *'Toolkits'* deve ser considerada. Como esta abordagem faz uso de componentes prontos e da arquitetura Centralizada, o desenvolvedor pode criar rapidamente novas aplicações colaborativas.

Por ser a primeira abordagem apresentada para o desenvolvimento de aplicações colaborativas, os *toolkits* se tornaram populares, sendo utilizados para a construção de diversos protótipos e provas de conceito.

A abordagem *'SCT - 1º geração'* permite o uso colaborativo de, virtualmente, qualquer aplicação, pois esta abordagem encasula as aplicações existentes agindo como uma plataforma operacional. Como nenhuma modificação no código fonte é sugerida por esta abordagem, o desenvolvedor deve apenas configurar a plataforma operacional

para que ela suporte o uso colaborativo da aplicação. Apesar de permitir o uso colaborativo de aplicações, Begole et al. [Begole et al. 1999] criticam esta abordagem afirmando que ela não suporta a colaboração adequadamente, argumentando que esta abordagem utiliza os recursos de rede de forma ineficiente, impondo um estilo de colaboração inflexível e altamente acoplado.

**Tabela 1. Comparações de abordagens.**

<b>Abordagem</b>	<b>Código Fonte</b>	<b>Requisito Tecnológico</b>	<b>Propósito</b>	<b>Arquitetura</b>	<b>Flexibilidade</b>
<i>Toolkits</i>	Não requer o código fonte	Depende da linguagem de programação	Criar novas aplicações	Centralizada	Muita
SCT - 1º geração	Não requer o código fonte	Nenhum	Promover colaboração	Centralizada	Pouca
SCT - 2º geração (ICT, ICT2 e ACT)	Requer API do sistema operacional	Requer camada de software	Promover colaboração	Centralizada ou Distribuída	Pouca
Adaptação Transparente	Requer API da aplicação	Requer camada de software	Promover colaboração	Replicada	Média
Substituição de Componentes	Requer o código fonte da aplicação	Depende da linguagem do componente	Criar novas aplicações e Promover colaboração	Replicada	Pouca
Mapeamento de Componentes	Requer o código fonte dos componentes da aplicação	Requer aplicação no estilo MVC	Promover colaboração	Híbrida	Muita

A abordagem ‘SCT - 2º geração’ também não sugere modificações no código fonte da aplicação. Contudo, para utilizar esta abordagem é necessário manipular a API do sistema operacional, que geralmente é complexa e extensa. Além do conhecimento necessário para manipular a API do sistema operacional, o desenvolvedor também deve conhecer as principais operações que o usuário pode realizar na aplicação, além dos detalhes específicos de cada sistema operacional que a aplicação for executada.

Devido a esta característica, o desenvolvedor se distancia do foco principal, que é tornar a aplicação colaborativa, ao se aprofundar na programação da API do sistema operacional. Além disso, o compartilhamento fornecido por esta abordagem se limita à replicação de eventos, não fornecendo recursos para o desenvolvedor implementar os demais aspectos funcionais comuns a aplicações colaborativas. Em contrapartida, esta abordagem é a única que permite a colaboração entre aplicações heterogêneas.

A próxima abordagem listada na Tabela 1, a abordagem ‘Adaptação Transparente’, possui como requisito tecnológico uma API da aplicação. Este tipo de requisito limita o uso desta abordagem a aplicações bem documentadas e que possuam uma API que permita a interceptação e reprodução de eventos, apresentado as mesmas

limitações da segunda geração de sistemas de colaboração transparente. O desenvolvedor que utilizar esta abordagem deve conhecer profundamente tanto a aplicação como sua API, uma vez que é por meio desta última que todas as funcionalidades colaborativas são implementadas.

Em comparação com as abordagens ‘*Toolkits*’, ‘SCT - 1º geração’, ‘SCT - 2º geração’, ‘Substituição de Componentes’ e ‘Mapeamento de Componentes’, a principal vantagem da abordagem ‘Adaptação Transparente’ é a separação da construção de um *groupware* da construção de uma camada de software, que propicia as funcionalidades colaborativas. Portanto, a abordagem ‘Adaptação Transparente’ implica no desenvolvimento de uma camada de software para capturar e replicar os eventos gerados pelas aplicações.

Com base na experiência do autor no uso da abordagem ‘Adaptação Transparente’, a alteração direta do código fonte da aplicação pode requerer menos recursos do que a utilização da API da aplicação. A aplicação da abordagem ‘Adaptação Transparente’ apresenta bons resultados, especialmente em aplicações comerciais complexas, como o editor de texto Word [Xia et al. 2004].

As abordagens ‘Substituição de Componentes’ e ‘Mapeamento de Componentes’ são as únicas abordagens que sugerem uma modificação direta na aplicação. Desta maneira, o desenvolvedor tem mais liberdade para implementar os aspectos colaborativos da maneira que desejar. Contudo, a manipulação direta do código fonte da aplicação pode acarretar em diversos problemas como, por exemplo, a inserção de novos defeitos na aplicação.

A abordagem ‘Substituição de Componentes’ sugere a troca de determinados componentes da aplicação por componentes colaborativos. Seguindo esta abordagem, o desenvolvedor substitui apenas partes específicas da aplicação para torná-la colaborativa. Além de demandar a modificação direta da aplicação, uma vez que um componente existente da aplicação será substituído por outro, esta abordagem é limitada pela oferta de componentes compatíveis com a tecnologia utilizada pela aplicação. Outra característica desta abordagem, descrita na Subseção 2.4, indica que apenas os requisitos relacionados à percepção podem ser implementados por esta abordagem.

A abordagem ‘Substituição de Componentes’ é abordagem que possui muitos pontos em comum com a abordagem ‘Mapeamento de Componentes’, em relação aos critérios utilizados nesta comparação. Contudo, o Mapeamento de Componentes se destaca em relação à Substituição de Componentes por não requerer recursos técnicos específicos, como uma determinada linguagem de programação ou ambiente de desenvolvimento, durante a implementação da abordagem. É interessante notar que, em certas aplicações, as abordagens ‘Substituição de Componentes’ e ‘Mapeamento de Componente’ podem ser utilizadas em conjunto, especialmente quando o mapeamento indicar que algum componente não pode ser modificado e deve ser substituído. Contudo, para que o desenvolvedor utilize estas duas abordagens em conjunto, é necessário que todos os requisitos de ambas as abordagens sejam satisfeitos.

De acordo com os dados da Tabela 1, o Mapeamento de Componentes deve ser utilizado no contexto das aplicações que possuem código fonte disponível, ou ao menos o código fonte dos principais componentes que constituem a aplicação. Devido a este valor para este critério, a maioria das aplicações candidatas para o uso do Mapeamento são as aplicações de código livre, onde um grupo de desenvolvedores disponibiliza o

código fonte para livre acesso. Também é importante notar que, em comparação com as outras abordagens, o Mapeamento de Componentes é o único que requer não apenas a estruturação da aplicação em componentes, mas também uma organização lógica dos mesmos, pois o estilo arquitetural MVC é um dos requisitos do mapeamento.

A abordagem ‘Mapeamento de Componentes’ apresenta um requisito adicional implícito no uso da arquitetura híbrida: a necessidade de um Servidor de Colaboração. Este Servidor de Colaboração deve ser desenvolvido a partir dos componentes do Modelo, o que implica em mais esforço de desenvolvimento.

Comparando o critério Disponibilidade do Código Fonte das abordagens ‘SCT - 1º geração’ ‘SCT - 2º geração’, ‘Adaptação Transparente’, ‘Substituição de Componentes’ e ‘Mapeamento de Componentes’, pode-se notar que as abordagens se dividem em dois grupos: o grupo de abordagens que não requer nenhum tipo de desenvolvimento, caracterizado pelo uso de plataformas operacionais, e o grupo das abordagens que requerem o desenvolvimento de infra-estrutura para a utilização colaborativa das aplicações, caracterizado pela necessidade de desenvolvimento.

O critério Requisito Tecnológico indica que as abordagens ‘SCT - 2º geração’, ‘Adaptação Transparente’, ‘Substituição de Componentes’ e ‘Mapeamento de Componentes’ requerem uma API do sistema operacional, uma API da aplicação, o código fonte da aplicação e o código fonte dos componentes da aplicação, respectivamente, além de recursos adicionais, com compilador, ferramentas de desenvolvimento etc. Estes requisitos nem sempre estão disponíveis ao desenvolvedor, limitando o uso destas abordagens a um conjunto específico de aplicações.

Com relação às arquiteturas utilizadas pelas abordagens da Tabela 1, Suthers [Suthers 2001] indica que a arquitetura Centralizada utiliza ineficientemente os recursos de rede, pois esta arquitetura transmite a informação completa de visualização e eventos da interface pela rede. Ainda segundo Suthers, a arquitetura Replicada possui a desvantagem de ser naturalmente mais adequada para aplicações WYSIWIS restritas e não WYSIWIS relaxadas. Já a arquitetura Distribuída é criticada por depender muito da rede, tornando-se inutilizável quando os serviços de rede estão indisponíveis. Suthers também critica a arquitetura Híbrida, argumentando que sua implementação é mais complexa do que as demais.

Sob o ponto de vista da Flexibilidade, as abordagens ‘SCT - 1º geração’ e ‘SCT - 2º geração’ apresentam pouca capacidade para fornecer requisitos colaborativos, pois elas não modificam a aplicação. De forma análoga, a abordagem ‘Substituição de Componentes’ limita-se apenas ao requisito percepção, pois esta abordagem sugere a substituição apenas dos componentes da interface gráfica.

A abordagem ‘Adaptação Transparente’ recebeu o valor ‘Média’ para o critério Flexibilidade, pois apesar de não modificar a aplicação, pode-se contar com os recursos da API para estender algumas partes da aplicação e implementar alguns requisitos colaborativos. As abordagens ‘Mapeamento de Componentes’ e ‘*Toolkits*’ possuem muita flexibilidade para implementar novos requisitos, devido à sua estratégia de criar novas aplicações a partir do zero ou modificar diretamente o código fonte.

Todas as abordagens comparadas, com exceção dos *toolkits*, permitem que aplicações não colaborativas possam apresentar recursos colaborativos, possibilitando a reutilização de aplicações existentes em um contexto colaborativo. Esta contribuição é

relevante, pois possibilita associar em uma mesma solução as ferramentas já existentes, adequadas à tarefa e provavelmente familiares aos participantes, com as funcionalidades colaborativas necessárias para realizar a tarefa em ambientes distribuídos colaborativos.

#### 4. Conclusões e Comentários Finais

Este artigo apresentou os critérios e uma comparação de abordagens disponíveis para implementar requisitos colaborativos em aplicações não colaborativas. A partir da comparação de abordagens apresentada neste trabalho, os desenvolvedores podem compreender melhor qual tipo de abordagem é mais recomendada, em termos dos critérios apresentados, na implementação dos requisitos colaborativos desejados, auxiliando os desenvolvedores na escolha da abordagem a ser utilizada

As abordagens ‘*Toolkits*’, ‘SCT - 1º geração’, ‘SCT - 2º geração (ICT, ICT2 e ACT)’, ‘Adaptação Transparente’, ‘Substituição de Componentes’, e ‘Mapeamento de Componentes’ foram apresentadas de acordo com o contexto na qual elas devem ser utilizadas. Os critérios ‘Código Fonte’, ‘Requisito Tecnológico’, ‘Propósito’, ‘Arquitetura’ e ‘Flexibilidade’ foram utilizados para a comparação, de acordo com o estudo bibliográfico e a experiência prática dos autores no uso das abordagens.

Para uma tomada de decisão efetiva, além dos critérios técnicos, devem-se considerar também os critérios gerenciais. Um critério gerencial importante que deve ser incluído na escolha final da abordagem é o esforço de análise, projeto, implementação e testes, se possível em termos quantitativos, como estimativas de Homens-Hora, ou eventualmente qualitativos. Esse tipo de critério gerencial tem valores específicos resultantes de um conjunto de atributos e valores da organização de desenvolvimento tais como: prontidão e habilidades dos membros da equipe, técnicas, ferramentas e processos adotados. Apesar dos critérios gerenciais serem importantes na decisão de qual abordagem adotar, a elaboração desses critérios gerencial foge ao escopo deste artigo. De qualquer forma, os autores julgam que critérios gerenciais na prática devam ser considerados na decisão final sobre escolha da abordagem.

Com a comparação apresentada neste trabalho, os desenvolvedores de aplicações colaborativas podem contar com um apoio para a escolha de uma determinada abordagem. Tendo em vista que a implementação de requisitos colaborativos apresenta vários desafios, contar com critérios de comparação pode facilitar o processo de adoção de uma determinada abordagem, incentivando a comunidade de desenvolvedores a modificar aplicações existentes para suportar funcionalidades colaborativas e, conseqüentemente, aumentando a quantidade de aplicações colaborativas nos diversos domínios de conhecimento.

**Agradecimentos.** Os autores gostariam de agradecer aos revisores anônimos que contribuíram para a melhoria deste trabalho.

#### 5. Referências

- Abdel-Wahab, H., Feit, M. A. (1991) “XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration”, em: Proceedings of the IEEE Tricomm, Chapel Hill, Carolina do Norte, E.U.A., p.159-167.
- Begole, J. C. A., Rosson, M. B., Shaffer, C. A. (1999) “Flexible collaboration transparency: supporting worker independence in replicated application sharing

- systems”, em: ACM Transactions on Computer-Human Interaction, volume 6, número 2, Junho de 1999, p.95-132.
- Ellis, C., Gibbs, S. J, Rein, G. L. (1991) “Groupware: some issues and experiences”, em: Communications of the ACM, volume 34, número 1, Janeiro de 1991, p.38-58.
- Garnkel, D., Welti B. C., Yip, T. W. (1994) “SharedX: A tool for real-time collaboration”, em: HP Journal, volume 45, número 2, Abril de 1994, p.23-36.
- Greenberg, S., Roseman, M (1998) Groupware toolkits for synchronous work, John Wiley & Sons, Nova York, primeira edição.
- Gutwin, C., Greenberg, S. (2002) “A Descriptive Framework of Workspace Awareness for Real-Time Groupware”, em: Computer-Supported Cooperative Work, volume 11, número 3, Julho de 2002, p.411- 446.
- Hill, J., Gutwin, C. (2004) “The MAUI Toolkit: Groupware Widgets for Group Awareness”, em: Computer Supported Cooperative Work, volume 13, números 5-6, Dezembro de 2004, p.539-571.
- Li, D., Li, R. (2002) “Transparent sharing: interoperation of heterogeneous single-user applications”, em: Proceedings of the 8th ACM Conference on Computer Supported Cooperative Work (CSCW'02), New Orleans, Louisiana, E.U.A., p.246-255.
- Lu, J., Li, R., Li, D. (2004) “A State Difference Based Approach to Sharing Semi-Heterogeneous Single-User Editors”, em: Proceedings of the Sixth International Workshop on Collaborative Editing Systems (IWCES6) in CSCW'04, Chicago, Illinois, E.U.A., 2004.
- Mangan, M. A. S. (2006) Uma abordagem para o desenvolvimento de apoio à percepção em ambientes colaborativos de software Tese (Doutorado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- Microsoft. ”NetMeeting Home”, web site acessado em abril/2007: <http://www.microsoft.com/windows/netmeeting/>.
- Pichiliani, M. C., Hirata, C. M. (2006) “A Guide to map application components to support multi-user real-time collaboration”, em: Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCon 2006), Georgia, E.U.A., 2006.
- Schuckmann, C., Kirchner, L., Schümmer, J., Haake, J. M. (1996) “Designing object-oriented synchronous groupware with COAST”, em: Proceedings of the 3rd ACM Conference on Computer Supported Cooperative Work (CSCW'96), Nova York, E.U.A., p.30-38.
- Suthers D. (2001) “Architectures for Computer Supported Collaborative Learning”, em: Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001), New Orleans, Louisiana, E.U.A., p.6-8, 2001.
- Xia S., Sun D., Sun, C., Chen D., Shen H. (2004) “Leveraging Single-user Applications for Multi-user Collaboration: the CoWord Approach”, em: Proceedings of the 9th ACM Conference on Computer Supported Cooperative Work (CSCW'04), Chicago, Illinois, E.U.A., p.162-171.

# Inter- e Intra-relações entre Comunicação, Coordenação e Cooperação

Hugo Fuks<sup>1</sup>, Alberto Raposo<sup>1</sup>, Marco A. Gerosa<sup>2</sup>,  
Mariano Pimentel<sup>3</sup>, Denise Filippo<sup>1</sup> & Carlos J. P. Lucena<sup>1</sup>

<sup>1</sup>Departamento de Informática – PUC-Rio  
Rua Marquês de São Vicente, 225 RDC – Gávea  
22453-900 – Rio de Janeiro – RJ – Brasil

<sup>2</sup>Centro Universitário Vila Velha  
Rua Comissário José Dantas de Melo, 21 – Boa Vista  
29102-770 – Vila Velha – ES – Brasil

<sup>3</sup>Departamento de Informática Aplicada – UNIRIO  
Centro de Ciências Exatas e Tecnologia  
Avenida Pasteur 458 – Urca  
22290-240 – Rio de Janeiro – RJ – Brasil

hugo@inf.puc-rio.br, abraposo@tecgraf.puc-rio.br,  
gerosa@les.inf.puc-rio.br, pimentel@uniriotech.br,  
denise@les.inf.puc-rio.br, lucena@inf.puc-rio.br

**Abstract.** *Based on the 3C Collaboration Model, in this article is described the mapping of a variety of collaborations forms onto the inter-relations between communication, coordination and cooperation. It is also seen that, in virtue of the intra-relations between these dimensions, it is necessary to give computational support for these three collaboration dimensions independently of the groupware specific purpose. These inter- and intra-relations are considered in the Groupware Engineering development cycle.*

**Resumo.** *Baseado no modelo 3C de colaboração, neste artigo descreve-se como diversas formas de colaborar são mapeadas nos inter-relacionamentos entre comunicação, coordenação e cooperação. Também é visto, em virtude das intra-relações entre estas dimensões, como é necessário dar suporte computacional para estas três dimensões da colaboração no groupware independentemente do seu propósito específico. Estas inter- e intra-relações são consideradas no ciclo de desenvolvimento da Engenharia de Groupware.*

## 1. Introdução

Neste artigo, a colaboração é analisada a partir de 3 dimensões: comunicação, coordenação e cooperação. Estas 3 dimensões, que por sua vez estão inter e intra-relacionadas, são reconhecidas em diferentes formas de colaborar. A comunicação, ou conversação consiste na troca de mensagens e na negociação entre pessoas; a coordenação consiste no gerenciamento de pessoas, suas atividades e recursos; e a cooperação é a produção que ocorre no espaço compartilhado por estas pessoas. Este modelo de colaboração foi originalmente proposto por Ellis et al. [1991], com algumas

diferenças na terminologia: o que Ellis denomina “colaboração” é aqui caracterizado como cooperação. Neste artigo, colaboração é a inter e intra-relação entre comunicação, coordenação e cooperação.

O modelo 3C aparece na literatura para classificar os sistemas colaborativos, como em Borghoff & Schlichter [2000]. Entretanto, somente algumas poucas tentativas foram feitas para se usar este modelo no contexto da implementação destes sistemas. Um exemplo é o modelo usado pelo Clover, que define 3 classes de funcionalidades denominadas comunicação, coordenação e produção [Laurillau & Nigay, 2002, Calvary et al., 1997]. É similar ao modelo 3C em termos de especificação funcional de sistemas colaborativos dado que ambos lidam com as 3 classes de funcionalidades que devem ter suporte computacional num groupware.

O modelo 3C de colaboração é usado como base para o desenvolvimento do AulaNet, um ambiente de ensino e aprendizagem baseado na web [Fuks 2000]. Os serviços do AulaNet são divididos em de comunicação, de coordenação e de cooperação. Na versão 3.0 deste ambiente, os serviços foram desenvolvidos com o uso de uma arquitetura [Barreto 2006] em que os *framework* e componentes são baseados no modelo 3C [Gerosa 2006]. O modelo também é usado nesta abordagem para definir a sistemática de classificação para os componentes e serviços, alinhando assim a etapa de implementação (montagem de *groupware*) com outras etapas do processo que também fazem uso do modelo 3C: análise, requisitos e testes [Pimentel 2006].

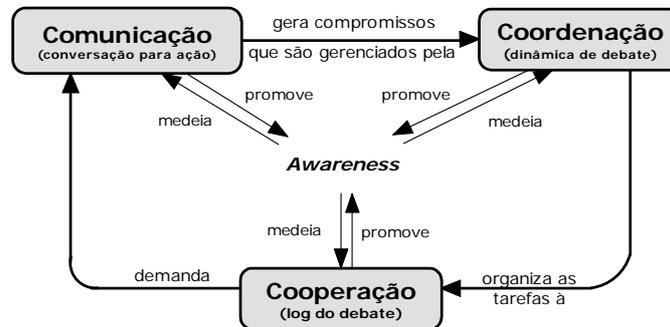
Devido à complexidade da interação de um grupo, o processo de testar sistemas colaborativos ainda não alcançou a sua maturidade. O modelo 3C também pode ajudar avaliadores a focar sua atenção em aspectos de comunicação, coordenação e cooperação, guiando a identificação de problemas de usabilidade. Uma abordagem de avaliação de sistema colaborativo baseado num modelo semelhante ao modelo 3C é apresentada em [Neale et al, 2004].

Na continuação deste artigo é investigado como estes 3Cs se inter-relacionam e são apresentadas 6 diferentes instanciações do modelo que, por sua vez, resultam em sistemas com propósitos distintos. Na seqüência, é apresentado como os 3Cs se intra-relacionam usando como exemplo uma ferramenta de bate-papo. Na conclusão do artigo, baseado na análise da interface do usuário e nos estudos sobre a visualização de formas da Psicologia Gestalt, indica-se como os indivíduos que usam sistemas implementados baseados neste modelo alternam entre atos de comunicação, de coordenação e de cooperação.

## **2. Os 3Cs Mapeados nos Sistemas Colaborativos: Inter-relações**

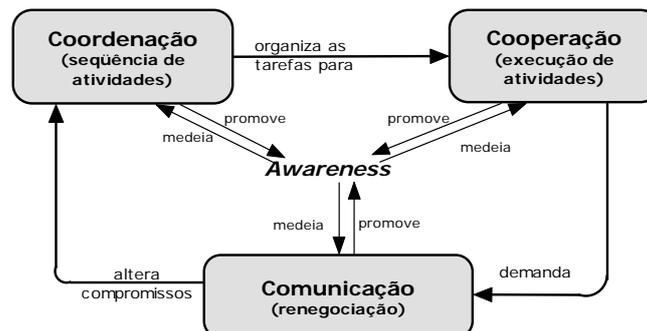
A primeira instanciação do modelo 3C tem seu foco no domínio do trabalho de grupo como representado na Figura 1, onde a conversação é voltada para a ação: enquanto se comunicam, as pessoas negociam e tomam decisões, e enquanto se coordenam, elas lidam com conflitos e organizam suas atividades de uma maneira que evita o desperdício de comunicação (como exemplifica a Perda de Co-texto [Pimentel et al., 2003]) e dos esforços de cooperação. A cooperação é a operação conjunta dos membros do grupo num espaço compartilhado, que executa as tarefas ao gerar e manipular objetos de cooperação. A necessidade de renegociar e tomar decisões sobre situações imprevistas que ocorrem durante a cooperação demanda uma nova rodada de

comunicação que, por sua vez, vai demandar coordenação para reorganizar as tarefas a serem executadas durante a cooperação. Este é o caso do serviço Debate do Ambiente AulaNet.



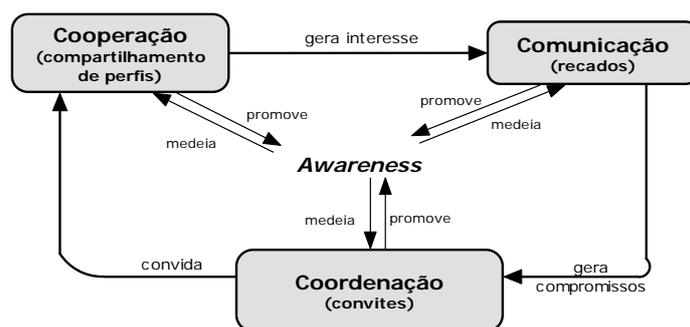
**Figure 1. A inter-relação dos 3Cs na conversação para ação**

A seqüência apresentada na Figura 2 tipifica um *workflow* adaptativo [Weber et al. 2004]. Ao executarem tarefas pré-articuladas por um *workflow*, membros do grupo sentem a necessidade de alterar o que fora previamente combinado. Um contra-exemplo é a linha de montagem industrial clássica, onde não se espera que os funcionários negociem a execução das próximas tarefas baseados nos resultados obtidos da execução das anteriores.



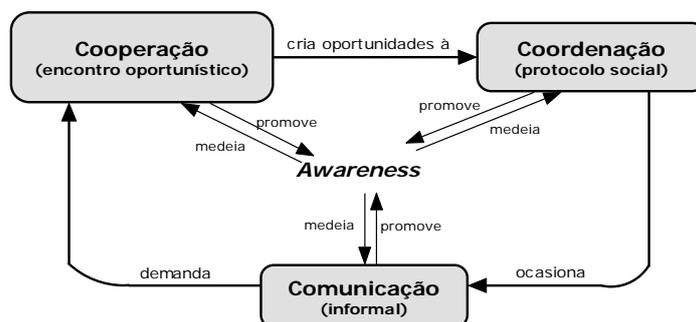
**Figura 2. A inter-relação dos 3Cs em sistemas de workflow adaptativo**

Os *websites* de relacionamento, como, por exemplo, o Orkut (<http://www.orkut.com>), têm o seu ciclo instanciado na Figura 3. Membros de uma comunidade colocam os seus perfis numa área comum a todos, que despertam a atenção e o interesse de alguns destes membros, que trocam mensagens, organizam festas namoram etc., e, eventualmente, convidam novos membros para se juntar à comunidade.



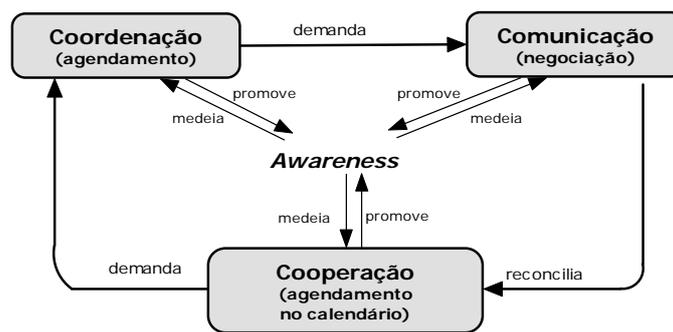
**Figura 3. A inter-relação dos 3Cs em sites de relacionamento**

Para Media Spaces [Mackay, 1999], que é a denominação atribuída aos espaços multimídia cujo objetivo é a comunicação informal entre pessoas, o modelo 3C é instanciado de acordo com a Figura 4. O Media Spaces é o espaço compartilhado propriamente dito através de vídeo distribuído como, por exemplo, o sistema RAVE desenvolvido na EuroPARC [Gaver et al. 1992]. Uma vez que ele é direcionado para a comunicação informal, seu objetivo principal é o de criar oportunidades para encontros casuais que são coordenados pelo protocolo social estabelecido, por exemplo, informando-se sobre a disponibilidade dos colegas remotos. Estes encontros propiciam conversas que ocorrem através do uso dos meios providos pelo próprio sistema ou disponíveis de outras formas, como por telefone.



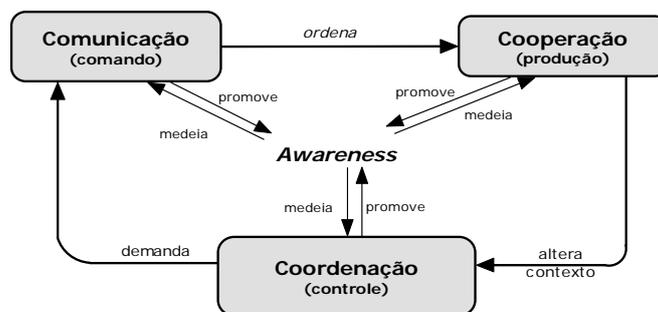
**Figura 4. A inter-relação dos 3Cs no Media Spaces**

A principal razão para se usar um calendário de família é o agendamento das atividades da família, que é o caso tratado pelo Family Calendar apresentado na Figura 5. Membros de uma família moderna têm uma variedade de interesses conflitantes que podem levar as combinações definidas na última noite a se tornarem inúteis na manhã seguinte. A fim de restaurar a coordenação da família, é necessária a negociação entre seus membros. “Este processo envolve ver o que já foi agendado, (...) e negociação de incumbências, horários e outras responsabilidades são necessárias” [Elliot & Carpendale, 2005]. A reconciliação obtida após a rodada de negociação é registrada no calendário compartilhado. Mas como a vida nunca pára, na manhã seguinte o ciclo começa de novo.



**Figura 5. A inter-relação dos 3Cs no Family Calendar**

Finalmente, é apresentada a instanciação das 3 dimensões do modelo 3C que caracteriza Comando e Controle (Figura 6). Aqui a coordenação é de fato uma supervisão, que vigia a produção do grupo e comanda mudanças. Um exemplo onde comando e controle estão colapsados é no remo onde o capitão observa o desempenho do grupo de remadores e grita cobrando mais empenho dos da direita ou esquerda.



**Figura 6. A inter-relação dos 3Cs no Comando e Controle**

Estes ciclos mostram a natureza iterativa da colaboração. Também evidenciam como os 3Cs interagem ‘entre’ si. Estas interações são mediadas pelas informações de *awareness* disponíveis no ambiente, que por sua vez, são influenciadas por cada dimensão da colaboração.

As inter-relações dos 3Cs apresentadas nesta seção ilustram como diferentes domínios de aplicação levam a diferentes combinações das dimensões da colaboração, que os respectivos sistemas colaborativos precisam de alguma forma refletir em seus projetos. Na próxima seção, é vista outra forma de relações entre os Cs. É mostrado que os 3Cs interagem ‘dentro’ si ou, em outras palavras, que existem os 3Cs dentro de um C. Também é apresentado o processo iterativo de desenvolvimento da Engenharia de Groupware baseada no modelo 3C de colaboração.

### 3. Dentro de cada C há os 3Cs: Intra-relações

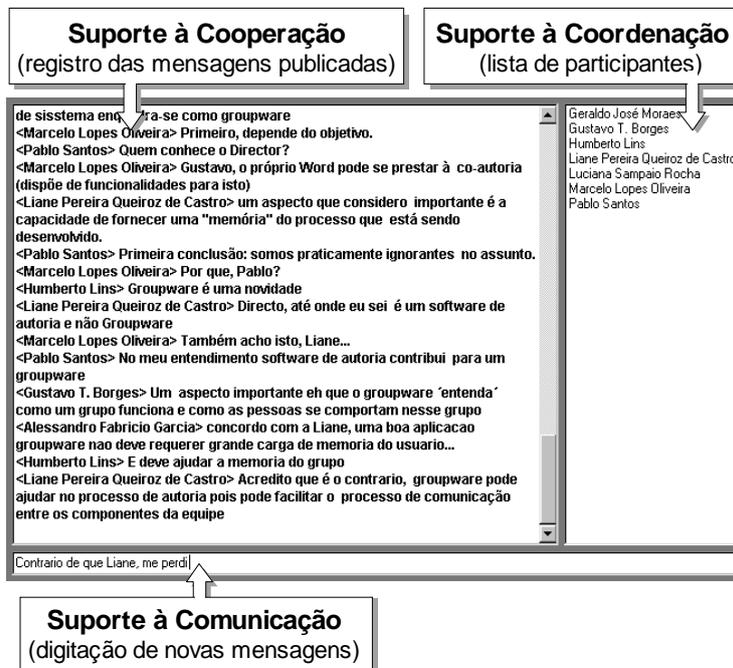
A Figura 7 apresenta como os exemplos da seção anterior se localizam dentro do triângulo 3C [Borghoff. & Schlichter, 2000 p. 125]. O primeiro C da inter-relação é o vértice do triângulo mais próximo do qual o sistema se localiza, por exemplo, no caso do *workflow* adaptativo, a coordenação. Porém, conforme evidenciado pelo triângulo, por mais direcionado para um determinado C que seja um *groupware*, este provavelmente lida com os outros 2Cs. 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. Pelo lado da cooperação, as mensagens trocadas entre membros de um projeto, contêm registros e informações que serão mineradas e recuperadas futuramente.



**Figura 7. Formas de colaborar no triângulo 3C**

Nesta seção, será analisada a diferenciação dos 3Cs dentro de cada C. Por exemplo, mesmo o bate-papo sendo uma ferramenta de comunicação cujo objetivo é possibilitar a troca de mensagens entre os membros de um grupo, também contém elementos de coordenação e cooperação que são necessários para organizar e documentar a comunicação (Figura 8).

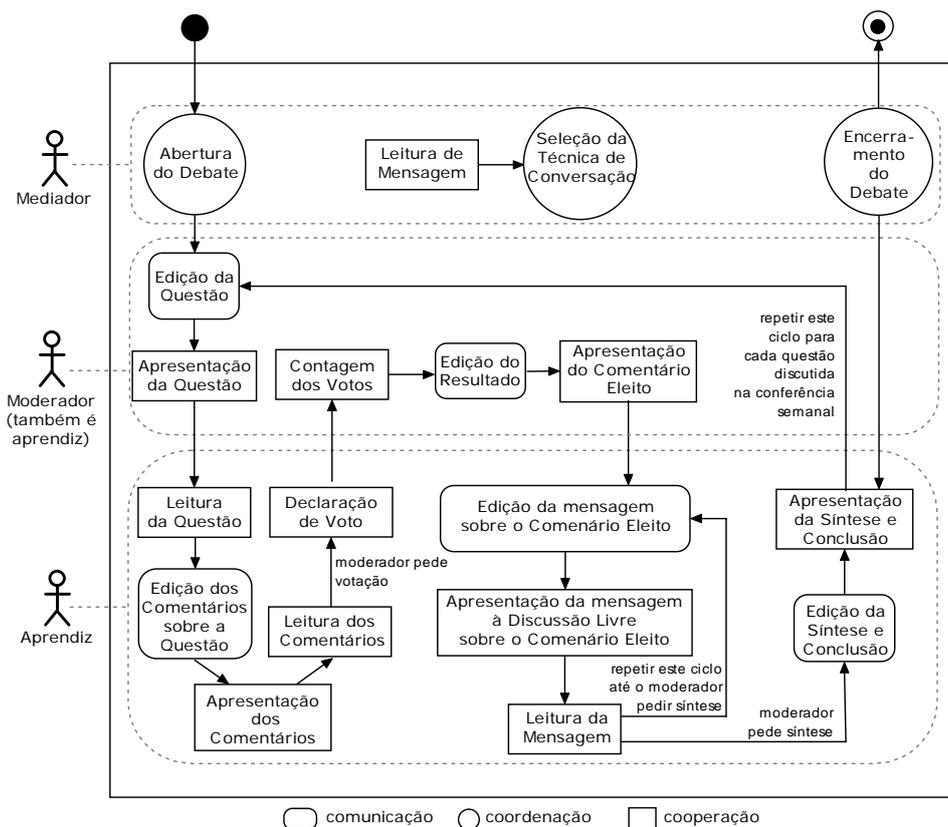


**Figura 8. Áreas de suporte às dimensões 3C de um sistema de bate-papo**

Ao se analisar uma ferramenta típica de bate-papo, são identificadas três áreas:

- a) uma área contendo um quadro para se digitar a mensagem, o que possibilita ao usuário se comunicar com os demais participantes, constituindo-se num suporte à comunicação;
- b) uma área contendo uma lista de participantes indicando os que estão conectados e disponíveis para a conversação, constituindo-se num suporte à coordenação; e
- c) uma área apresentando o registro das mensagens enviadas, constituindo-se num suporte à cooperação.

O curso Tecnologias da Educação Aplicadas à Educação (TIAE), atualmente na sua 17ª edição, que é ministrado *online* através do Ambiente AulaNet [Fuks et al. 2002] tem na sua fase inicial uma atividade colaborativa onde se debatem sincronamente os tópicos daquela semana. O serviço Debate oferece o devido suporte computacional para esta atividade. Originalmente, este serviço era baseado numa ferramenta similar ao sistema de bate-papo apresentado na Figura 8. No bate-papo ou *chat* típico, a única forma de se coordenar o debate é através do protocolo social existente, onde os aprendizes respeitam o moderador, também aprendiz, devido à presença (silenciosa) dos mediadores do curso. As dificuldades surgiram na 10ª edição do curso quando a dinâmica do debate mudou.

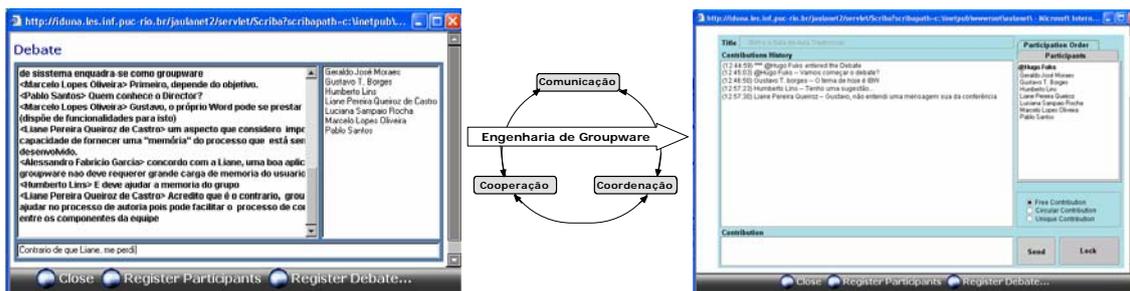


**Figura 9. Diagrama de atividades dos debates do curso TIAE**

A nova dinâmica do debate é apresentada na Figura 9. Nesta figura, círculos representam atos de coordenação, retângulos de cooperação e retângulos com bordas arredondadas de comunicação. A coordenação está a cargo dos mediadores e compreende a abertura e o encerramento do debate. Além destes dois momentos

extremos, durante o debate, os mediadores ao optarem por uma determinada técnica de conversação, deixam a conversação fluir livremente, de forma circular ou permitem o envio de uma única mensagem por aprendiz (usada na votação, por exemplo). Eles também podem bloquear o envio de mensagens dos aprendizes, inclusive do moderador, a qualquer instante [Rezende 2003]. A comunicação fica por conta dos aprendizes, ao editarem e enviarem mensagens. A cooperação, que é a produção conjunta dos aprendizes no espaço compartilhado, que no caso de um bate-papo, corresponde à leitura e à apresentação das mensagens (seminário, questões, votos etc.). Estas atividades, edição, submissão, seleção, votação etc. não são atômicas e poderiam por sua vez, ser subdivididas. Entretanto, por se tratarem de atividades individuais, fogem ao escopo desta investigação.

O que ocorre é que, quando a dinâmica do debate deixou de ser uma discussão livre dos tópicos e passou a ser organizada através de etapas bem definidas, como as da Figura 8, o protocolo social demonstrou-se insuficiente, e fez-se necessário adicionar novas funcionalidades ao serviço. Apesar de se tratar de um serviço de comunicação, as novas funcionalidades adicionadas, como por exemplo, mecanismos de controle de técnicas de conversação são de coordenação, dado que as alterações na dinâmica implicam em mudanças nesta dimensão. Esta adaptabilidade evidencia a força do modelo na evolução de um sistema colaborativo.



**Figura 10. Interfaces anteriores e atuais do serviço Debate do Ambiente AulaNet**

A Figura 10 apresenta a evolução das interfaces do serviço Debate resultante da nova versão do *Mediated Chat* [Pimentel et al. 2005] obtida através do processo de desenvolvimento da Engenharia de *Groupware* baseada no modelo 3C de colaboração [Fuks et al. 2005]. Esta engenharia não trata do desenvolvimento da interface usuário, mas esta reflete as novas funcionalidades do sistema colaborativo. O desenvolvimento em sucessivas versões, sempre através de prototipação [Brooks, 1975], é especialmente útil para os sistemas colaborativos uma vez que mudanças na ferramenta geram, por vezes, modificações imprevistas e indesejáveis na maneira do grupo trabalhar, sendo adequado reverem a solução implementada numa versão seguinte. O desenvolvimento de cada nova versão deve estar focado num C. Contudo, nem sempre é possível alterar apenas um único C por versão. Ao se mexer numa das dimensões da colaboração, é preciso estar ciente que os Cs são interdependentes, e ao se alterar um pode se fazer necessária a alteração de outro. O objetivo de se focar num único C é de fazer o projetista de groupware se questionar como o novo mecanismo implementado influencia a comunicação, a coordenação e a cooperação. Também não é necessário se restringir a resolução de um único problema por versão – há casos em que é adequado

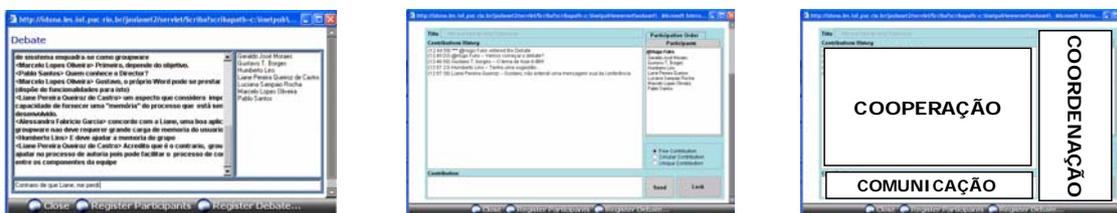
resolver dois ou mais problemas numa única versão, como no caso acima, onde algumas alterações também foram feitas no que tange à cooperação e à comunicação [Fuks et al. 2006].

Enquanto no *chat* típico somente se exibia a lista dos participantes presentes, no *Mediated Chat*, os nomes dos mediadores aparecem em destaque. Há também uma nova aba onde os participantes presentes são listados por ordem de chegada. Entretanto, a alteração que deu suporte computacional à nova dinâmica foi a implementação do mecanismo de técnicas de conversação. Através deste mecanismo, os mediadores definem se a conversação será livre, circular, contribuição única ou ainda, bloqueá-la para os aprendizes, mantendo-a liberada aos mediadores. Estas foram as modificações na coordenação. Quanto à comunicação, foi aumentado o número de caracteres máximo por mensagem e foi habilitado o pular de linha usando a tecla *ENTER*, o que levou a necessidade de um botão específico para disparar o envio da mensagem. Finalmente, na área de registro, que dá suporte à cooperação, todas as mensagens passaram a ser precedidas do seu *timestamp*, as mensagens do sistema foram precedidas de ‘\*\*\*’ e, herdou a diferenciação nos nomes dos mediadores implementada na lista de presença.

Nesta seção foi mostrado que, ao se entrar num C, é provável se deparar novamente com os 3Cs. Na investigação descrita neste artigo, esta análise se limita a um único nível. Na próxima seção conclui-se o artigo argumentando como o membro de um grupo alterna sua atenção entre os 3Cs disponíveis num groupware, para mostrar a adequação deste modelo à percepção humana.

#### 4. Conclusão: Adequação do Modelo 3C à Percepção Humana

Uma das primeiras lições que um desenvolvedor de *groupware* aprende é que, para desenvolver *groupware* de boa qualidade, é preciso entender de CSCW. Parte deste conhecimento é abordado na discussão relativa à interface do usuário. Observa-se na Figura 11 que, apesar das alterações, estas foram feitas de modo que as áreas que dão suporte a cada um dos C permaneçam praticamente estáveis: a cooperação compreende o retângulo acima à esquerda; a comunicação, o retângulo na parte inferior; e a coordenação, o retângulo vertical à direita. Não há espalhamento de elementos de um determinado C na área de outro C, e sobremaneira, estas áreas são visualizadas como blocos claramente distintos que compõe a interface do usuário.



**Figura 11. Blocos distintos para cada C na interface do usuário do serviço Debate**

Os agrupamentos dos mecanismos 3C em áreas bem definidas como apresentados na Figura 11 favorecem a percepção. A Psicologia Gestalt [Köhler, 1947] estuda a capacidade de percepção de formas que os nossos sentidos têm, especialmente no que diz respeito à visualização de figuras completas, preferencialmente fechadas, em vez de um conjunto de linhas e curvas. Cabe ressaltar que percepção difere de *awareness* (no modelo 3C de colaboração é usado *awareness*). Informação de

*awareness* [Gutwin & Greenberg, 2004] é provida pelo sistema colaborativo para os participantes terem ciência (*awareness*) de informações relevantes para o trabalho. Informação de *awareness* tem que ser provida pelo sistema, pois os sentidos humanos não são capazes de captar eventos em locais remotos (como a chegada de um colega numa sala distante) nem conseguem acompanhar a sobrecarga dos eventos ocorridos ao longo de uma sessão de trabalho.

Outro princípio da Psicologia Gestalt usado nesta investigação é *Figure and Ground* [Rubin, 1958]. Em certas imagens, percebe-se que uma parte emerge (torna-se figura) e, enquanto nossa atenção se concentra nela, o resto da imagem desaparece (constituindo-se o fundo). Quando parte deste fundo emerge tornando-se figura, a figura anterior passa a fazer parte do fundo. McLuhan [1988] estende este princípio para toda a estrutura da percepção e da consciência: em todas as situações há uma área de atenção (*figure*) e de outra área maior ainda de desatenção (*ground*), e que estas áreas interagem continuamente entre si através de uma fronteira comum que serve para definir ambas de forma simultânea – “Cada nova figura desloca a anterior de volta para o fundo”. No caso do serviço Debate, entre o terminar de uma etapa e o iniciar de outra (como, por exemplo, terminar a digitação de uma mensagem e consultar o registro das mensagens publicadas), ocorre o deslocamento da atenção: o que era figura (comunicação) vira fundo e parte do fundo anterior vira a figura atual (cooperação). McLuhan afirma que “na ordem natural das coisas, o fundo vem primeiro, e a figura emerge depois”, em outras palavras, sem fundo não há figura, o que reforça a convicção de que são necessários os 3Cs para de fato haver colaboração no seu sentido mais amplo.

Finalmente, todas as instâncias do modelo 3C (inter-) aqui descritas são implementáveis com as técnicas de desenvolvimento baseado em componentes (intra-) abordadas na introdução deste artigo.

## **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 José Pereira de Lucena nº 300091/2003-6, Hugo Fuks nº 301917/2005-1, Marco Aurélio Gerosa nº 383719/06-2. Carlos José Pereira de Lucena e Hugo Fuks também recebem bolsas individuais do projeto Cientistas do Nosso Estado da FAPERJ. Denise Filippo recebe bolsa CCPG/VRAC da PUC-Rio.

## **Referências**

- Barreto, C.G. (2006): Agregando Frameworks de Infra-Estrutura em uma Arquitetura Baseada em Componentes: Um Estudo de Caso no Ambiente AulaNet. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, 14 de março.
- Borghoff, U.M. & Schlichter, J.H. (2000): Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, USA.
- Brooks Jr., F.P. (1975): Plan to Throw One Away. In: The Mythical Man-Month – Essays on Software Engineering, chap. 11, pp. 115-123. Addison-Wesley.

- Calvary, G., Coutaz, J., Nigay, L. (1997): From Single-User Architectural Design to PAC\*: a Generic Software Architectural Model for CSCW. Conference on Human Factors in Computing Systems (CHI'97), pp 242-249.
- Elliot, K. & Carpendale, S. (2005): Awareness and Coordination: A Calendar for Families. Technical Report 2005-791-22, Department of Computer Science, University of Calgary, Calgary, Alberta, CANADA, T2N 1N4, May.
- Ellis, C.A., Gibbs, S.J. & Rein, G.L. (1991): Groupware - Some Issues and Experiences. Communications of the ACM, v. 34, N. 1, pp. 38-58.
- Ellis, C.A. & Wainer, J. (1994): A Conceptual Model of Groupware, In T. Malone (ed): Conference on Computer-Supported Cooperative Work (CSCW), pp. 79-88.
- Fuks, H. (2000): Groupware Technologies for Education in AulaNet. Computer Applications in Engineering Education, NY, v. 8 N. 3 & 4 December. Online ISSN: 1099-0542 Print ISSN: 1061-3773 Wiley-InterScience, pp 170-177.
- Fuks, H., Gerosa, M.A. & Lucena, C.J.P. (2002): The Development and Application of Distance Learning on the Internet. Open Learning - The Journal of Open and Distance Learning, v. 17, N. 1, ISSN 0268-0513, pp 23-38.
- Fuks, H., Raposo, A.B., Gerosa, M.A. & Lucena, C.J.P. (2005): Applying the 3C Model to Groupware Development. International Journal of Cooperative Information Systems (IJCIS), v.14, n.2-3, Jun-Sep, World Scientific, pp. 299-328.
- Fuks, H., Pimentel, M. & Lucena, C.J.P. (2006): R-U-Typing-2-Me? Evolving a chat tool to increase understanding in learning. International Journal of Computer-Supported Collaborative Learning, v.1, N.1. ISSN: 1556-1607 (Paper) 1556-1615 (Online). Springer: Março. pp. 117-142.
- Gaver, W., Moran, T., MacLean, A., Löfstrand, L., Dourish, P., Carter, K. and Buxton, W. (1992): Realizing a video environment: EuroPARC's RAVE system. In *Proceedings of Human Factors in Computing Systems, CHI '92* (Monterey, CA), pages 27-35. ACM Press, New York,.
- Gerosa, M.A. (2006): Desenvolvimento de Groupware Componentizado com Base no Modelo 3C de Colaboração. Tese de Doutorado, Departamento de Informática, PUC-Rio, 16 de março.
- Gutwin, C., and Greenberg, S. (2004): The Importance of Awareness for Team Cognition in Distributed Collaboration. In E. Salas and S. M. Fiore (Editors) *Team Cognition: Understanding the Factors that Drive Process and Performance*, pp. 177-201, Washington:APA Press.
- Köhler, W. (1947): Gestalt Psychology: An Introduction to New Concepts in Modern Psychology, ISBN 0-87140-218-1, Liveright Publishing Company, NY, re-impresso 1992.
- Laurillau, Y. & Nigay, L. (2002): Clover architecture for groupware, Conference on Computer-Supported Cooperative Work (CSCW), pp. 236 - 245
- Mackay, W. E. (1999): Media Spaces: environments for Informal Multimedia Interaction. In M. Beaudouin-Lafon (ed): Computer Supported Co-operative Work (Trends in Software: 7). John Wiley & Sons, England, pp. 55-82.

- McLuhan, M. & McLuhan, E. (1988): *Laws of Media: The New Science*, ISBN 0-8020-7715-3, Toronto: University of Toronto Press.
- Neale, D. C., Carroll, J. M. & Rosson, M. B. (2004): Evaluating Computer-Supported Cooperative Work: Models and Frameworks. Conference on Computer-Supported Cooperative Work (CSCW), pp. 112-121.
- Pimentel, M., Fuks, H. & Lucena, C.J.P. (2003): Co-text Loss in Textual Chat Tools. 4th International and Interdisciplinary Conference on Modeling and Using Context - CONTEXT 2003, LNAI 2680, Stanford, CA, USA, June, pp 483-490.
- Pimentel, M., Fuks, H. & Lucena, C.J.P. (2005): Mediated Chat Development Process: Avoiding Chat Confusion on Educational Debates. Computer Supported Collaborative Learning, Taiwan, ISBN 0805857826, Lawrence Erlbaum Associates, pp. 499-503, July.
- Pimentel, M. (2006): RUP-3C-Groupware: um processo de desenvolvimento de groupware baseado no Modelo 3C de Colaboração. Tese de Doutorado, Departamento de Informática, PUC-Rio, 22 de março.
- Rezende, J.L. (2003): Aplicando Técnicas de Comunicação para a Facilitação de Debates no Ambiente AulaNet. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, março.
- Rubin, E. (1958): Figure and ground. In Readings in perception (DC Beardslee and M Wertheimer, Eds). Van Nostrand: Princeton, NJ.
- Weber, B., Wild, W. & Breu, R. (2004): CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning. LNCS v. 3155, p. 434-438, ISBN 978-3-540-22882-0, Springer Berlin / Heidelberg.

# Um Modelo para Gestão de Conhecimento Baseado em Contexto

Vanessa Tavares Nunes<sup>1</sup>, Flavia Maria Santoro<sup>2</sup>, Marcos R.S. Borges<sup>1</sup>

<sup>1</sup>NCE-IM- Universidade Federal do Rio de Janeiro  
Caixa Postal 68.530- CEP 21941-590 - Rio de Janeiro – RJ

<sup>2</sup>DIA – Universidade Federal do Estado do Rio de Janeiro – UNIRIO  
Av Pasteur 458, Urca, RJ 22290-240

vanunes@gmail.com, flavia.santoro@uniriotec.br, mborges@nce.ufrj.br

***Abstract.** Knowledge Management (KM) has become a prominent subject for organizations, but often characterization and treatment of the information that flows in a well-defined work process are muddled. We argue that context is the fundamental information resource for better understanding and carrying on activities and interactions. Therefore, we propose a model to support the cycle of creating and dealing with information about process activities and interaction performed, focusing on their context. The goal of this paper is to present the model and also to specify mechanisms that can be used to capture context information within the KM environment.*

***Resumo.** A Gestão de Conhecimento (GC) se tornou um assunto proeminente para as organizações, porém muitas vezes as caracterizações e o tratamento da informação que flui durante a execução de um processo de trabalho são confusas. Contexto é um recurso de informação fundamental para o melhor entendimento e execução das atividades e das interações. Propõe-se um modelo de GC, integrado ao ambiente de trabalho, que apóia a criação, manipulação e reutilização de informações contextuais relacionadas às atividades em execução. O objetivo deste artigo é apresentar o modelo e mais especificamente padrões para orientação ao desenvolvimento e acoplamento de mecanismos de captura destas informações.*

## 1. Introdução

O termo Gestão do Conhecimento (GC) tem adquirido muito espaço nos últimos anos dentro das organizações. De acordo com Concklin (1997), “conhecimento é o recurso chave das organizações”. As instituições reconhecem que o desenvolvimento e o gerenciamento eficientes do conhecimento organizacional são fatores decisivos para o sucesso e a qualidade dos produtos/serviços (Nonaka e Takeushi, 1997, Davenport e Prusak, 1998).

Neste cenário, em que o conhecimento se torna peça importante para o aprendizado organizacional, é necessário obter não só o conhecimento proveniente dos artefatos gerados durante a execução das atividades de processos de trabalho, mas também as situações e caracterizações ocorridas durante a sua realização. Estas situações e caracterizações se traduzem em questões como: Como se deu a interação? O que foi discutido? Por que uma ação foi executada dessa forma? Quais habilidades eram requeridas para executar a ação e quais eram possuídas pelos executores?

Porém, há muitos casos em que a documentação, bem como o tratamento das informações, são deixadas de lado ou armazenadas de forma caótica, não permitindo aos profissionais absorverem os fatos e circunstâncias ocorridas, relacionadas à informação necessária para executar seu trabalho. O conhecimento já explorado em situações anteriores e semelhantes pode antecipar problemas, resolver questões mais rapidamente, descartar maus exemplos e sugerir bons exemplos.

Adiciona-se a esta situação o fato de que a ausência de apoio adequado à captura, armazenamento e reutilização do conhecimento pode resultar em informações truncadas, sem sentido ou desconexas, com pouca ou nenhuma ligação entre si e as atividades que estão sendo executadas. Pesquisas (Brézillon, 2002) reconhecem que a captura, o gerenciamento e a disponibilização do conhecimento devem ser considerados juntamente com o contexto ao qual esse conhecimento foi capturado, gerenciado e disponibilizado. O conhecimento contextual reside nas experiências de cada profissional, em cada artefato e também nas atividades, condições, fatos e situações que ocorrem na execução do trabalho.

As pessoas utilizam informações de contexto sem se dar conta disto. A importância do conhecimento contextualizado se deve ao fato deste possuir a capacidade de prover maior significado à atividade, fatos, artefatos gerados e decisões tomadas. Além disso, o conhecimento contextual, quando tratado em um modelo de gestão de conhecimento, atua como um filtro que apóia na descoberta de informações comparando o contexto atual com aqueles disponíveis na memória organizacional.

Se a percepção sobre o conhecimento contextual for ampliada, maior será o apoio à execução das atividades em um processo de trabalho. Para ampliar esta percepção, é necessário explicitar o conhecimento contextual, representá-lo de maneira uniforme, centralizá-lo e torná-lo acessível a todos os envolvidos.

O objetivo deste artigo é apresentar um modelo para gestão de conhecimento baseado em contexto que trata o ciclo de criação, armazenamento e reutilização de conhecimento contextual, e que abrange as questões de representação do conhecimento, captura, armazenamento, comparação e apresentação no cenário da execução de uma atividade em um processo de trabalho. O modelo de gestão de conhecimento visa estabelecer uma memória organizacional com o resultado da execução das atividades dos processos de trabalho com o seu contexto, através dos quais resultados foram alcançados.

## **2. Gestão de Conhecimento e Contexto**

De acordo com Davenport e Prusak (1998), a maioria dos projetos relacionados à gestão do conhecimento possui um dos seguintes alvos: tornar o conhecimento visível e mostrar o papel do conhecimento na organização; desenvolver uma cultura de conhecimento através do encorajamento e da agregação de hábitos como compartilhamento de conhecimento e procura e oferta de conhecimento de forma pro ativa; criar uma infra-estrutura de conhecimento, e não somente um sistema tecnológico; e criar uma rede de conexões entre pessoas provendo espaço, ferramentas e estimulando a interação e a colaboração.

Desta forma, a implantação da gestão do conhecimento deve cobrir todos os estágios formando um ciclo de criação e compartilhamento. Algumas pesquisas (Te'eni e

Weinberger, 2000) sugerem que pouco ainda foi investido no estágio de captura do conhecimento. Vários esforços hoje em dia se concentram em desenvolver repositórios de dados que armazenam documentos e dados ou bancos de dados de discussão nos quais os participantes recordam suas próprias experiências em relação a algum problema ou respondem algum comentário, como as Comunidades Virtuais (Bosua e Scheepers, 2002 e Hildret et al, 2000) e o uso de técnicas de group storytelling (Perret et al., 2004).

Criar um ambiente real de trabalho onde exista a colaboração e o compartilhamento de informações requer mais do que somente codificar o conhecimento, armazená-lo em sistemas de informação e prover acesso e distribuição. Isto é também um problema de facilitar a interação, comunicação e entendimento mútuo.

A colaboração entre as pessoas no sentido de compartilhar conhecimento ocorre quando elas estão genuinamente interessadas em trocar experiências para aprender continuamente e aprimorar seu trabalho. Com as constantes mudanças que ocorrem nos negócios, seja por questões de economia, tecnologia ou competição, os profissionais precisam aprender com os outros para obter uma percepção mais abrangente e completa das situações cotidianas, e dessa forma serem passíveis de inovar.

A criação, armazenamento e compartilhamento de conhecimento nas organizações, no entanto, enfrenta algumas barreiras (Concklin, 1997). A primeira delas é fazer com que o conhecimento tácito e informal seja explicitado, o que envolve entendimento do conhecimento de indivíduos com experiências, especialidades e formas de trabalhar diferentes, e que, portanto, precisam criar um entendimento comum. Outra barreira é a falta de preservação do raciocínio existente por trás da confecção dos registros que são armazenados na MO, desprovendo-os de significado, o que faz com percam o sentido quando o cenário se modifica. Por fim, a última trata dos aspectos de relevância do conhecimento que é manipulado pela organização.

De fato, o maior problema existente na GC não reside na falta de informação e sim na sua qualidade, conteúdo, organização e raciocínio. Um dos desafios pelos quais a organizações têm se defrontado está relacionado à relevância da informação para a organização, mais precisamente para os diversos contextos em que cada grupo de trabalho dentro da organização está situado. E ainda, a informação tende a perder a relevância e, por conseqüência, seu valor, através do tempo. Em um futuro muito próximo irão existir incontáveis volumes de conhecimento para serem acessados, o que tornará cada vez mais difícil encontrar aqueles cujo conteúdo seja relevante para o problema que está sendo tratado em um determinado momento.

O raciocínio, as discussões e ações que levam à produção ou alteração de uma informação ou conhecimento no momento da execução de um processo de trabalho o posicionam num determinado tempo e espaço. Uma forma de tornar a informação relevante é associá-la ao contexto na qual foi produzida.

Araujo e Brézillon (2005) reconhecem que a captura, o gerenciamento e a disponibilização do conhecimento tácito devem ser considerados juntamente com o contexto ao qual esse conhecimento foi capturado. O conhecimento contextual reside nas

experiências de cada profissional, em cada artefato e também nas atividades, condições, fatos e situações que ocorrem durante a execução de um trabalho.

Contexto pode ser definido como “as circunstâncias em que um evento ocorre”. Quando se trata da comunicação entre pessoas, contexto representa a história de tudo que ocorreu durante um período de tempo, o estágio geral de conhecimento dos agentes participantes em um dado momento e um pequeno conjunto de expectativas existentes naquele momento.

Enfim, para o total entendimento de ações e eventos, é necessário ter acesso às informações contextuais relevantes que envolvem essas circunstâncias. Todo este conhecimento não é uma parte das ações que serão executadas ou dos eventos que ocorrem, mas irá subsidiar de forma mais eficaz e eficiente a execução de uma ação ou de uma interpretação do evento sem intervir nele explicitamente (Brézillon, 2002).

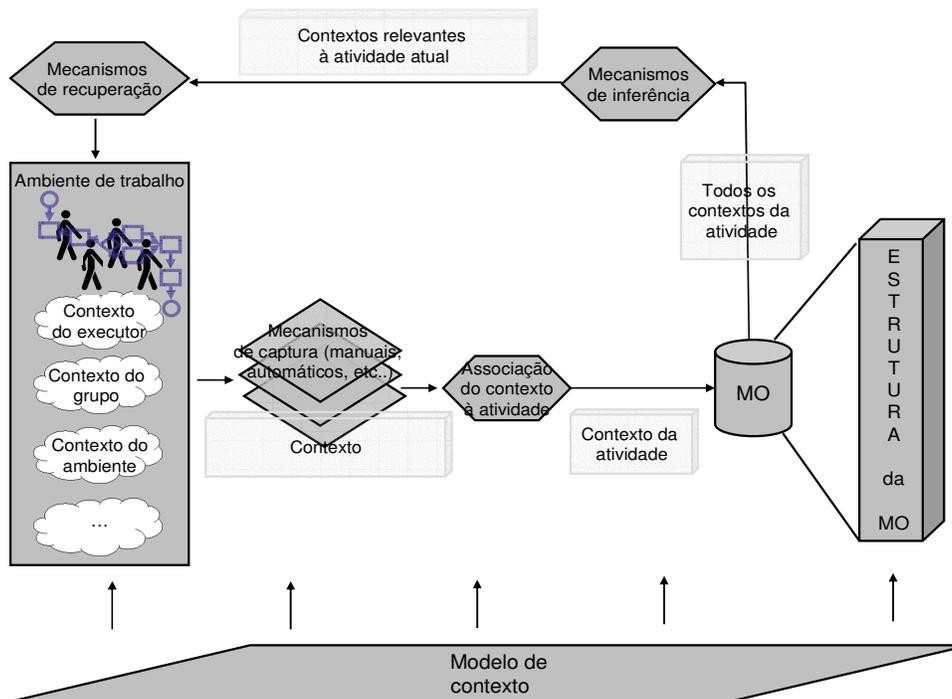
O contexto em um processo de trabalho tem uma natureza dinâmica, onde novos eventos surgem e novas decisões são tomadas, alterando o curso de sua execução. Uma organização que não associa informações de contexto às atividades que executa e artefatos que gera tem em sua memória organizacional um imenso conjunto de documentos com pouca ou nenhuma conexão entre eles. Como esta memória não possui contexto associado, ou seja, um significado, ela é muitas vezes ignorada como um recurso de informações.

A gestão das informações de contexto, quando implementada em um ambiente computacional, pode atuar como um filtro que define, em um determinado momento, qual conhecimento deve ser levado em consideração no auxílio à execução de tarefas. Por isto, entende-se que capturar e disponibilizar as circunstâncias que resultaram na tomada de uma decisão ou na criação de um artefato permitirão uma reutilização com maior qualidade e utilidade do conhecimento gerado na própria organização em situações e contextos futuros.

### **3. Um Modelo de Gestão de Conhecimento baseado em Contexto**

Para atender às questões levantadas, propõe-se um modelo para o ciclo de criação, armazenamento e reutilização de conhecimento contextual apresentado da Figura 1, e que abrange as questões de representação, acesso, armazenamento, comparação e recuperação no cenário da execução de uma atividade em um processo de trabalho. O modelo proposto define componentes que permeiam o ambiente de trabalho dos profissionais na organização, incorporando a gestão de conhecimento (Nunes, 2007).

A proposta tem como foco a atividade do processo de trabalho e o seu objetivo é criar um ambiente onde os executores destas atividades possam obter ajuda na realização das mesmas. A MO é organizada em uma estrutura definida que mapeia os contextos existentes nos objetos de conhecimento armazenados, como documentos, bancos de dados, endereços eletrônicos, indivíduos, grupos, etc. Sendo a atividade o foco, no ambiente de trabalho supõe-se a existência de processos definidos e institucionalizados onde diversos agentes atuam em conjunto para produzir um resultado e atingir um objetivo em comum. Este ambiente é composto de agentes, lugares físicos e uma infra-estrutura de apoio à execução das atividades dos processos.



**Figura 1 – Modelo de GC baseado em contexto integrado à execução de uma atividade**

O ciclo de transformação do contexto (Figura 1) é contemplado e o modelo é discutido sob três aspectos importantes que englobam toda a máquina de Gestão de Contexto integrado à GC na execução da uma atividade, detalhados a seguir: Representação das informações de contexto; Captura e armazenamento das informações de contexto; Recuperação e apresentação de contextos existentes.

### 3.1. Representação das informações contextuais

Para capturar e usar informações de contexto é necessário especificar quais informações de contexto serão tratadas no âmbito da organização e depois representá-las em um formato compreensível e aceitável por todos. O elemento “Modelo de contexto” estabelece uma representação explícita do conhecimento que está sendo capturado.

A determinação das informações necessárias para a composição do contexto não é uma atividade trivial. Vários tipos de informações contribuem com o contexto, e a relevância de cada pedaço de informação depende da situação em questão. Fatores externos, como artefatos produzidos, discussões, mensagens trocadas e ações realizadas são relativamente simples de serem identificadas e capturadas. Por outro lado, aspectos internos do indivíduo – interesses, objetivos, experiências, etc, são de difícil tratamento.

Uma representação formal do contexto ainda permite que mecanismos de raciocínio lógico possam ser usados para checar a consistência das informações de contexto, possam realizar comparações com outros contextos e possam inferir novas informações complexas a partir dos contextos existentes. Algumas pesquisas apontam para formas de representação de contexto tais como em Rosa et al.(2003) e Vieira et al. (2005).

O modelo proposto nesta pesquisa é uma ontologia que descreve os elementos de contexto associados às atividades no processo de trabalho (descrito em Nunes et al., 2006) e reproduzido na Figura 2.

### **3.2. Captura e armazenamento das informações contextuais**

O primeiro passo no sentido de compor a memória organizacional é começar pela captura do conhecimento que flui durante a execução das atividades dentro do ambiente de trabalho. Informações de contexto devem ser coletadas (elemento “Mecanismos de captura” da Figura 1) durante a sua execução ou imediatamente após o seu término. Os mecanismos para capturar o contexto (do executor, do grupo, do ambiente, etc) devem procurar a informação que está disponível. A captura pode ser puramente humana onde o próprio profissional que participa da atividade registra o seu contexto. Outras informações se encontram disponíveis no próprio ambiente ou fazem parte do resultado da execução da atividade propriamente dita, o que requer um mecanismo que automaticamente identifique-as e capture-as de acordo com o modelo de representação.

Outro ponto a ser considerado é a forma como esses mecanismos se acoplam ao ambiente de trabalho e ainda como eles influenciam na execução das atividades dentro do processo. Para que as informações de contexto possam ser recuperadas, elas devem estar sinalizando uma unidade de tempo bem definida que marca a circunstância em que foram capturadas. Desta forma as informações contextuais capturadas devem ser associadas à atividade (elemento “Associação” da Figura 1) que está sendo executada assim como informações provenientes da MO podem ser associadas de acordo com os objetivos, características e necessidades que cada atividade possui.

### **3.3. Recuperação e apresentação de contexto**

O resultado da captura, associação à atividade e armazenamento das informações contextuais gera o contexto da atividade atual que realimenta a MO (elementos “MO” e “Estruturas da MO” da Figura 1), introduzindo conhecimentos, novos ou não, na organização que poderão ser consultados quando da execução de atividades semelhantes. Sendo o objetivo principal a reutilização de conhecimento, é necessária a criação de mecanismos de inferência (elemento “Mecanismos de inferência” da Figura 1) que sejam capazes de recuperar contextos existentes relacionados a uma atividade passada que possam auxiliar na execução da atividade atual.

Para cada atividade específica, seu conhecimento contextual pode ser inferido através da MO baseado nas características das atividades atuais definidas em um conjunto de regras de contexto. Esses mecanismos, considerados automáticos, podem se apresentar (elemento “Mecanismos de recuperação”) na forma de mecanismos de percepção que através das regras de contexto existentes elegem aquele conhecimento considerado útil para o conjunto atividade e executor. De outra forma, a recuperação de informações de contextos de atividades passadas pode ser manual, permitindo com o que o próprio profissional possa recuperar o conhecimento desejado. A captura de novos conhecimentos bem como a interação direta do indivíduo com a interface para sua recuperação pode gerar novas regras de contexto, contribuindo para que os mecanismos de percepção atuem no sentido de realizar uma seleção cada vez mais otimizada e proveitosa.



**Sensores físicos:** mecanismos, em sua maioria, eletro-mecânicos que monitoram e captam informações sobre o ambiente físico em que os profissionais atuam e a interação entre eles.

**Agentes semi-automáticos:** componentes lógicos que requerem certo nível de interação entre o profissional e o sistema, ou seja, dependem parcialmente de algum tipo de comando externo para serem acionados.

Mecanismos humanos são utilizados para capturar informações contextuais relacionadas à atividade e aos seus agentes. O registro humano destes conhecimentos tenta capturar os passos seguidos pelos agentes e também o seu ponto de vista em relação às dificuldades, artefatos produzidos e conhecimento adquirido e utilizado neste meio. Um exemplo de mecanismo humano é a apresentação de uma interface acoplada ao software utilizado durante a execução de uma atividade. Os profissionais que estão executando a atividade registram as informações de contexto solicitadas através desta interface.

Os agentes automáticos são utilizados para capturar informações contextuais de qualquer um dos grupos de informação. São componentes ou módulos de software que buscam também na MO informações já existentes sobre os profissionais e sobre as atividades. Um exemplo de agente automático é um componente de software construído e acoplado ao software utilizado durante a execução de uma atividade. Ele continuamente checa por informações utilizadas ou registradas pelos executores, associando-as à atividade.

Os sensores físicos permitem a captura de informações de contextos referentes ao mundo físico e às interações que ocorrem entre os profissionais que se encontram no mesmo ambiente de trabalho. As informações capturadas devem ser tratadas por serem armazenadas em mídias complexas e de difícil busca por informações. Um exemplo de sensor físico são câmeras de vídeos acopladas ao ambiente físico dos profissionais.

Os agentes semi-automáticos podem ser utilizados para capturar informações relacionadas às atividades, às pessoas, aos agentes e as interações. São aplicações, que a partir de regras pré-estabelecidas ou que surgem do aprendizado proveniente das execuções de uma determinada atividade, são capazes de capturar informações de contexto relacionadas à situação, mas que necessitem ser validadas ou acionadas opcionalmente pelo profissional. Um exemplo de mecanismo semi-automático é um componente de software acoplado a uma ferramenta de troca de mensagens. Este componente somente captura as mensagens trocadas mediante o acionamento dos participantes.

#### **4.1. Padrões: acoplamento dos mecanismos ao ambiente de trabalho**

Para tornar a proposta operacional, é necessário especificar estes mecanismos em termos de funcionalidades e definir como serão inseridos no ambiente de trabalho. De forma a especificar os mecanismos que serão inseridos no ambiente, um conjunto de padrões é proposto. Os padrões visam facilitar o processo de seleção e desenvolvimento destes de acordo com o ambiente de trabalho em que se encontram os profissionais que fazem parte dele.

Em termos gerais os padrões apresentados seguem os seguintes passos: definir quais informações devem ser capturadas (relevância das informações para o processo em questão); definir como será feita a associação das informações à cada atividade do processo

de trabalho; criar o componente de software que pode ser: uma aplicação isolada (conectada a MO), uma aplicação integrada a outra(s) no ambiente ou interfaces com dispositivos externos conectadas a MO. Quatro padrões foram elaborados.

**Interromper para Registrar:** Atividades podem ou não ser apoiadas por software. É necessário coletar o contexto manualmente através de mecanismos humanos.

**Registrar sem interromper:** Atividades são necessariamente apoiadas por groupware. O contexto está disponível na MO e é automaticamente capturado.

**Registrar através de sensor físico:** Atividades, ou parte delas, usam sensores físicos que capturam automaticamente o contexto sobre o ambiente.

**Acionar registro:** Atividades são necessariamente apoiadas por groupware. O contexto é capturado automaticamente mediante a permissão de um ou mais executores.

### Exemplo de padrão:

<b>Nome:</b> Interromper para registrar
<b>Ambiente corrente de trabalho:</b> Grupos de pessoas trabalham em suas atividades correntes que estão caracterizadas em processos conhecidos e bem definidos. As fronteiras entre as atividades são conhecidas (pontos de início e fim, entradas e saídas). Algumas atividades são apoiadas por software, mas não necessariamente todas elas.
<b>Problema:</b> As pessoas não param a execução de suas tarefas de trabalho para prover informação contextual sobre a atividade atual ou outra já finalizada.
<p><b>Solução</b></p> <p>Introduza uma nova atividade no processo, uma interrupção no curso normal, que é exatamente o momento em que a informação contextual será fornecida. A informação contextual será provida ao ambiente através da apresentação da interface de um mecanismo de captura.</p> <p>Dois pontos devem ser considerados: (i) os participantes da atividade devem ser avisados sobre a interrupção no curso normal de seu processo de trabalho; (ii) a informação deve ser explicitamente solicitada aos participantes através da interface do mecanismo de captura ou através da definição desta atividade no processo de trabalho.</p> <p>Siga o padrão para introduzir o componente no ambiente de trabalho:</p> <ul style="list-style-type: none"> <li>- Cheque a necessidade de capturar cada informação através dos grupos de classes do modelo de contexto. Identifique as informações de contexto específicas do domínio e customize o modelo de contexto para o negócio em questão acrescentando novas classes e atributos e excluindo aqueles que não são necessários.</li> <li>- Identifique no processo onde cada informação deve ser capturada e introduza nestes pontos as atividades onde cada informação deve ser capturada, ou seja, cada mecanismo humano será inserido.</li> <li>- Construa um mecanismo de captura para cada atividade inserida no processo de acordo com a metodologia própria do desenvolvedor. Estes mecanismos são aplicações de software que ficarão embutidos no ambiente de trabalho, seja acoplado a algum sistema que já apóia a execução do processo seja em uma aplicação a parte. Para uma aplicação a parte considere a inserção de um mecanismo que indique aos usuários o melhor momento para usá-lo.</li> </ul> <p>Exemplo: Durante um processo de realização de reunião, após a discussão de cada tópico da agenda, a secretária registra suas impressões acerca dos motivos pelos quais se chegou a um resultado específico.</p> <p>A funcionalidade da aplicação é descrita no caso de Uso “Registrar contexto manualmente”.</p>
<p><b>Caso de Uso</b></p> <p>O caso de uso inicia quando os Agentes (pode existir mais de um participante na atividade) de processo estão diante de um ponto marcado durante a execução do processo de trabalho:</p> <p>Pré-condições: Agente(s) de Processo iniciou(aram) a execução da atividade do processo</p> <p>Fluxo Normal:</p> <ol style="list-style-type: none"> <li>1. O sistema apresenta a interface do mecanismo e solicita as informações de contexto especificadas anteriormente.</li> <li>2. O(s) Agente(s) de Processo provê(em) as informações solicitadas.</li> <li>3. O sistema registra as informações na MO de acordo com o modelo de ontologia.</li> </ol> <p>Pós-Condições:</p> <ul style="list-style-type: none"> <li>- Informação de contexto registrada.</li> </ul> <p>Fluxo Alternativo:</p> <ol style="list-style-type: none"> <li>1. O sistema apresenta a interface e solicita as informações de contexto sobre a atividade</li> <li>2. O(s) Agente(s) de Processo cancela(m) o registro e continua executando a atividade.</li> </ol>

O estudo e análise cautelosos dos ambientes de trabalho onde estes “componentes” são inseridos é o ponto de partida. As características das atividades definem que tipo de informação dever ser capturada e associada e como estas podem ser obtidas. Seguindo os padrões propostos, os mecanismos são especificados e podem ser construídos e implementados em ambiente reais.

Na seção seguinte será apresentado um cenário de forma a ilustrar a construção dos mecanismos de captura de conhecimento contextual e a sua inserção em um ambiente de trabalho dando forma ao modelo de gestão de conhecimento proposto. O cenário descrito foi baseado em estudo realizado em uma organização real, porém são ocultados nomes e identificações, e o processo foi também alterado para descaracterizá-lo do verdadeiro.

## **5. Avaliação da Proposta: Um Cenário de Uso do Modelo**

A PInfoTech é uma organização de informática especializada em desenvolvimento de soluções de software corporativas. Atua em diversos setores de negócios desenvolvendo soluções específicas de acordo com a necessidade do cliente. A organização se encontra há um bom tempo no mercado e possui um número considerável de projetos já finalizados. A partir das necessidades do cliente e conjuntamente com ele, os analistas de requisitos elicitam e especificam os requisitos. A PInfoTech se encarrega do desenvolvimento do software proposto, realizando testes de validação do sistema e de verificação junto ao cliente e implantando o software na empresa contratante.

### **5.1. O Processo de Trabalho**

O trabalho do analista de requisitos é iniciado a partir de uma solicitação de novo software ou de manutenção evolutiva feita por um usuário ou grupo de usuários. Em primeiro lugar o analista verifica os processos de negócio relacionados e analisa junto ao cliente a pertinência da demanda. Caso seja considerada válida no contexto do processo de negócio, o desenvolvimento é iniciado.

Sendo a demanda pertinente, o analista levanta os requisitos de negócio o que corresponde a identificar junto aos clientes quais são os objetivos do sistema, como o sistema se encaixa no contexto das necessidades do negócio, e como deverá ser a utilização do sistema no dia a dia da organização cliente.

A seguir, o analista de requisitos analisa e documenta os requisitos de negócio levantados. Ele explora os relacionamentos entre requisitos, examina consistência, omissão, ambigüidade, e prioriza os requisitos com base nas necessidades dos clientes. Para este trabalho, a ferramenta RequisitePro (REQPRO, 2006) é utilizada, com o objetivo de documentar os requisitos levantados e realizar a sua gerência.

O analista de requisitos e os clientes chegam então a um entendimento e a um acordo sobre o delineamento do escopo do projeto. Podem ser necessárias algumas revisões nos requisitos de negócio e retornos aos clientes para que ocorra este entendimento.

### **5.2. Gestão do Conhecimento Coletivo neste Ambiente**

Este cenário procurou focar na captura de contexto de atividades cujo histórico tenha maior possibilidade de ser utilizado como apoio na execução da atividade atual.

Sendo assim, no momento da validação dos requisitos e da revisão dos mesmos caso não sejam validados, é necessário capturar o contexto de suas execuções, tanto do ponto de vista da equipe de desenvolvimento quanto do cliente. Se o analista de requisitos já souber o que aconteceu em revisões anteriores, os tipos de problemas ocorridos e as abordagens de validação que não deram certo, no momento em que ele for fazer o planejamento da validação e da revisão atual, ele pode ter informações úteis para decidir por estratégias que tenham uma probabilidade maior de sucesso.

O primeiro passo no sentido de introduzir mecanismos de captura de contexto foi verificar qual ou quais mecanismos seriam necessários para o domínio e o processo selecionados. Para esta situação foram escolhidos os tipos de mecanismos: humano e agente semi-automático, e os padrões “Interromper para registrar” e “Acionar registro” cujos passos foram seguidos.

De acordo com os dois padrões descritos, os passos seguintes foram realizados para implementar os mecanismos de captura.

Cheque a necessidade de capturar cada informação através dos grupos de classes do modelo de contexto. Identifique as informações de contexto específicas do domínio e customize o modelo de contexto para o negócio em questão acrescentando novas classes e atributos e excluindo aqueles que não são necessários.

Foram definidas quais informações do domínio deveriam ser capturadas e para tanto a ontologia de contexto foi utilizada como base. Tendo a engenharia de requisitos como domínio considerado, a ontologia foi especializada de forma a comportar as suas especialidades. Foi introduzida a classe Requisito como uma especialização de Artefato que comporta os requisitos trabalhados nas duas atividades que estão sendo tratadas.

Identifique no processo onde cada informação deve ser capturada e introduza nestes pontos as atividades onde cada informação deve ser capturada, ou seja, cada mecanismo humano será inserido. Identifique no processo onde cada informação será capturada e analise o impacto (de tempo, espaço, custo) da inserção de mecanismos automáticos de captura na arquitetura do sistema que apóia a execução do processo.

Para os dois padrões, a segunda etapa realizada se constituiu na alteração do processo para a inclusão de atividades onde a informação contextual deve ser fornecida. Duas atividades foram inseridas como forma de introduzir uma interrupção no processo para permitir a inclusão das informações de contexto pertinentes. Além disso, paralelamente às atividades de validação e revisão dos requisitos de negócio passa a existir a captura semi-automática de informações contextuais.

### **5.3. Implementação dos Mecanismos de Captura**

Seguindo o padrão, os próximos passos estão voltados para a construção dos mecanismos de captura semi-automática (primeira atividade) e humana (segunda atividade) e seus acoplamentos ao ambiente de trabalho. Para o mecanismo de captura semi-automático levaram-se em consideração os seguintes aspectos abordados pelo padrão:

Analise também o modelo de dados do ambiente computacional (ferramentas de apoio ao processo) onde o mecanismo será inserido. Será necessário criar uma associação entre as informações contextuais e os metadados do modelo de dados para que o agente seja capaz de realizar a captura

O gerente de requisitos somente precisa iniciar uma reunião de validação na interface por trás da qual foi implementado o mecanismo de captura. A recuperação das

informações especificadas durante a execução da reunião acontece automaticamente, sem ônus tanto para o Gerente de Requisitos quanto para o Analista de Requisitos. Quanto ao segundo aspecto, foi analisada a forma como o RequisitePro armazena os requisitos em sua base de dados de forma a associar uma reunião de validação à um projeto no RequisitePro.

Para o mecanismo de captura humano, foi seguido o passo:

Construa um mecanismo de captura para cada atividade inserida no processo de acordo com a metodologia própria do desenvolvedor. Estes mecanismos são aplicações de software que ficarão embutidos no ambiente de trabalho, seja acoplado a algum sistema que já apóia a execução do processo seja em um sistema a parte. Para uma aplicação a parte considere a inserção de um mecanismo que indique aos usuários, e os lembre, o melhor momento para usá-lo.

Para promover a captura destas informações, foi necessário introduzir mecanismos de captura no ambiente onde este processo é executado. A atividade “Validar requisitos de negócio” já utilizava a aplicação de software RequisitePro. Para apoiar a captura das informações de contexto foi necessário integrar uma aplicação a esta. Para isto, foi desenvolvida uma interface de captura de contexto para ler os requisitos existentes em um projeto criado no RequisitePro. Esta interface foi desenvolvida tendo como unidade de tratamento a realização de uma reunião de validação ou uma reunião de revisão de requisitos não validados.

Construa a interface de configuração e habilitação dos mecanismos. Nesta interface os executores, de acordo com regras definidas pela organização, podem ter a possibilidade de selecionar quais classes e quais atributos destas classes terão informações de contexto sendo capturadas e podem também, especificar o momento (temporal ou lógico) em que tais mecanismos irão iniciar a captura. Esta interface pode ser acoplada ao sistema que apóia a execução do processo ou ao ambiente computacional dos executores.

Neste caso, a captura das informações de contexto pelo agente semi-automático é realizada durante a realização da atividade. Portanto, na interface desenvolvida, quando o registro de uma nova reunião acontece, a informações referentes à atividade de validação são capturadas e registradas automaticamente para aquela instância do processo. Para que o mecanismo semi-automático reconheça o início da captura de contexto, o analista de requisitos deve acionar o registro desta nova reunião.

A partir das informações de contexto selecionadas para estas atividades, foram estabelecidas três etapas de desenvolvimento da interface: 1) O desenvolvimento de mecanismos automáticos de captura das informações de contexto que já residem na memória organizacional; 2) A integração com a ferramenta RequisitePro para a captura dos requisitos registrados durante a atividade “Analisar e documentar requisitos de negócio”; 3) O desenvolvimento da interface de captura humana das informações de contexto que decorrem do momento da execução da reunião. Além disso, é necessário associar em cada requisito registrado na aplicação RequisitePro as informações de contexto referentes a reunião de validação em questão. A interface desenvolvida incorporou as três etapas seguindo o caso de uso especificado nos padrões (Figura 3).

Ainda para este ambiente foi desenvolvida uma ferramenta que agrega um fórum de discussão onde o Analista, o Gerente de Requisitos e o Cliente podem registrar suas impressões, sentimentos e dúvidas.

**Virtual Listbox**  
Learning Project - Use Cases

**1**

Reunião: Reunião de Validação

PROCESSO: Levantamento de requisitos de negócio  
 META: Aumentar a aderência entre os processos de negócio e os requisitos que apoiam os mesmos.  
 ATIVIDADE: Validar requisitos de negócio  
 DESCRIÇÃO DA ATIVIDADE: O cliente, juntamente com o analista de requisitos, valida os requisitos de negócio apoiado pelos processos de negócio da organização. Essa validação pode ser realizada tanto localmente como remotamente.  
 OBJETIVO ESPERADO: Validação de todos os requisitos de negócio para que seja possível prosseguir para o desenvolvimento do sistema.  
 PRE-REQUISITOS: Levantamento realizado e requisitos de negócio aderentes as atividades de cada processo.  
 RESPONSÁVEL: Analista de requisito  
 HIERARQUIA DE RESPONSABILIDADE: Gerente de requisitos  
 FUNÇÃO: Levantar, documentar e validar os requisitos

**2**

Data/Hora de Início: 15/11/2006 10:30  
 Prazo (Data/hora): 15/11/2006 12:00  
 Data/Hora de Fim: 15/11/2006 12:40

Local: sala oval  
 Localização: sala oval sala 385 sala de reuniao

Sistema Computacional: Net Meeting  
 Sistema Computacional selecionado: Net Meeting Windows XP comunicação remota

Responsáveis: Vanessa Nunes, Flavia Santoro, Fernanda Baião

Objetivo alcançado: A reunião de validação está acontecendo

Requisitos de Negócio:

UC1.3 INVENTORY NOT AVAILABLE |A| WAREHOUSE SYSTEM RESPONDS, some of the order (or the entire order) is not available, and the Ware  
 UC1.4 INVALID INFORMATION |A| BF WAREHOUSE SYSTEM RESPONDS, the Warehouse System finds that some of the information received is  
 UC1.5 NO RESPONSE |A| BF WAREHOUSE SYSTEM RESPONDS, there is no response from the Warehouse System; the system flags the order  
 UC1.6 ORDER PLACED |A| An order has been placed and the Warehouse System has already been notified.  
 UC1.7 SUCCESS |U|on successful completion the system has provided all necessary information to the Warehouse System, and the Shopper's orde  
 UC2 Check Order Status  
 UC2.1 Registered ClassicsCD.com Shoppers can view the status of their order(s) by entering their CustomerID and Password. They can then read a st  
 UC2.2 BEGIN |I|The use case starts when the Shopper chooses to check on the status of a previous order. |IDENTIFY SHOPPER |I|The system req  
 UC2.3 TRACK PACKAGES |A| BF VIEW ORDER DETAILS, the order has already shipped; the Shopper chooses to view tracking information for the  
 UC2.4 SHOPPER ALREADY LOGGED ON |A| BF IDENTIFY SHOPPER, the Shopper who has already logged on the system bypasses BF IDENTIF

3

Validar requisito selecionado  
 Recusar requisito selecionado  
 Limpar requisito selecionado

Salvar Cancelar

**Figura 3. Tela de registro de contexto de uma reunião**

## 6. Conclusões e Trabalhos Futuros

O modelo de ambiente de gestão de conhecimento e os padrões de aplicação propostos tem como objetivo guiar o desenvolvedor na identificação, construção e acoplamento de mecanismos de captura ao ambiente onde um processo é executado. Para demonstrar a construção de mecanismos de captura incorporados ao ambiente de trabalho foi realizada uma avaliação da proposta.

Os resultados do exercício de captura do conhecimento contextual são conjuntos de informações que combinam as representações de contextos tácitos e explícitos existentes na organização. Capturar as informações contextuais em uma estrutura uniforme, conforme a ontologia proposta e utilizando mecanismos de captura que estejam em sincronia com esta direção mostrou que o conhecimento pode ser mais facilmente compartilhado por todos os envolvidos no projeto, promovendo a colaboração.

Existe, porém um custo associado à implantação e manutenção de um ambiente para gestão de conhecimento contextual e a implantação de uma cultura de compartilhamento e reuso de informações, que deve ser considerado. Problemas políticos, de natureza competitiva podem influenciar na forma e tipos de informação que serão armazenadas assim como o medo existente nos profissionais de uma organização que ao transmitirem o conhecimento sobre o negócio que dominam, se tornem descartáveis.

Para que ocorra a captura e conseqüente reuso do conhecimento, os profissionais devem adquirir um novo pensamento: eles não são mais receptores passivos do

conhecimento, eles são pesquisadores, construtores e comunicadores ativos do conhecimento. O conhecimento não deve ser criado apenas por gerentes. Ele deve ser construído de forma colaborativa, no contexto do trabalho. Por isso é importante que tanto o conhecimento formal quanto o conhecimento informal sejam coletados e compartilhados, para que a organização aprenda coletivamente. Embora cada projeto de trabalho seja único, experiências similares podem auxiliar os profissionais a executarem suas atividades e a resolverem problemas atuais existentes no seu ambiente de trabalho.

## **Referências Bibliográficas**

- Araujo, R. M., Brézillon, P., Reinforcing Shared Context to Improve Collaboration, *Revue d'Intelligence Artificielle*, França, v. 19, n. 3, p. 537-556, 2005.
- Bosua, R., Scheepers, R., Towards web support for tacit knowledge exchanges. In: *Proceedings of the Information Systems Research Seminar In Scandinavia (IRIS25)*. Bautahøj, Denmark., 2002.
- Brézillon P. Making context explicit in communicating objects. *Communicating Objects*, C. Kintzig, G. Poulain, G. Privat, P-N. Favennec (Eds.), Hermes Science Editions, Lavoisier, 2002.
- Concklin, E.J.: Capturing Organizational Memory. In: Coleman, D. (ed.) *Proceeding of GroupWare*. Morgan Kaufman, San Mateo. CA, 133-137, 1992.
- Davenport, T.H, Prusak, L., *Working Knowledge: How Organizations Manage What They Know* , Harvard Business School Press, Boston, 1998.
- Hildret, P., Kimble C., Wright P., Communities of Practice in the Distributed International Environment. *Journal of Knowledge Management*. 4. (1) pp27-38, 2000.
- Nonaka, I. Takeushi, H., *Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação*, Rio de Janeiro, Ed. Campus, 1997.
- Nunes, V.T., Santoro, F. M., Borges, M. R. S., Context in Knowledge-Intensive Collaborative Work, in: *10th International Conference on Computer Supported Cooperative Work in Design (CSCWD'2006)*, Nanjing, China, 2006.
- Nunes, V.T., *Um Modelo de Gestão de Conhecimento baseado em Contexto*, Dissertação de Mestrado, NCE-IM/UFRJ, 2007.
- Perret, R., Borges, M.R.S., Santoro, F.M., Applying group storytelling in knowledge management. In: *CRIWG 2004*. San Carlos, Costa Rica, 2004.
- REQPRO, IBM RequisitePro, Disponível em: <http://www-306.ibm.com/software/awdtools/reqpro/>, Acesso em: 10/11/2006.
- Rosa, M.G.P. Borges, M.R.S. Santoro, F.M., A Conceptual Framework for Analyzing the Use of Context in Groupware, *LNCS*, Berlin, Germany, Vol. 2806, p. 300-313, 2003.
- Te'eni D., Weinberger H., System Development of Organizational Memory: a Literature Survey. In: *Proceedings of the 8th European Conference on Information Systems*, Austria, Hansen, H. R., Bicheler M., and Mahrer H. (eds.), vol. 1, pp. 219-227, 2000.
- Vieira V., Tedesco P., Salgado A. C., Towards and Ontology for Context Representation in Groupware, In: *Proceedings of the 11th International Workshop (CRIWG 2005)*, Porto de Galinhas, Brazil, 2005.

# Sistema de Co-Navegação com Suporte a Áudio-Conferência

Christopher V. Lima<sup>1</sup>, Roberto Willrich<sup>1</sup>,  
Roberta Lima-Gomes<sup>2</sup>, Guillermo de Jesús Hoyos-Rivera<sup>3</sup>

<sup>1</sup>Depto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)  
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brazil

<sup>2</sup>Depto de Informática – Universidade Federal do Espírito Santo (UFES)  
Av. Fernando Ferrari, S/N, 29060-970 – Vitória – ES – Brazil

<sup>3</sup>Universidad Veracruzana  
Sebastián Camacho # 5, 91000 – Xalapa, Veracruz – México  
{cviana,willrich}@inf.ufsc.br, rgomes@inf.ufes.br, ghoyos@uv.mx

**Abstract.** *Co-browsing allows several users synchronizing their browsing activities, from different computers. The CoLab tool provides simple and powerful synchronization facilities for dynamically organizing co-browsing sessions. Users are allowed to create independent workgroups in the same co-browsing session. However, for effectively co-browsing, it is important to employ a communication tool in order to allow users to discuss about co-browsed subjects. This paper presents a proposal for integrating CoLab with an audio-conference controller. In order to integrate these two applications, we adopted the integration environment for collaborative applications.*

**Resumo.** *A co-navegação permite a um grupo de usuários navegarem em páginas Web de forma sincronizada, cada um em seu próprio computador. CoLab é uma ferramenta de co-navegação permitindo o gerenciamento de sessões de co-navegação onde os membros podem dinamicamente se auto-organizar em grupos de trabalho independentes. Para possibilitar uma co-navegação efetiva, os membros das sessões de co-navegação devem ter acesso a uma ferramenta de comunicação, permitindo a discussão do assunto co-navegado. Este artigo apresenta um sistema de co-navegação composto de CoLab e de um Controlador de Conferências. Para esta integração foi adotado o ambiente de integração de aplicações colaborativas LEICA.*

## 1. Introdução

A navegação colaborativa, ou co-navegação, tem como objetivo básico estender a navegação Web tradicional, permitindo que vários usuários, cada um em seu próprio computador, naveguem em conjunto, de forma sincronizada. A co-navegação adiciona uma nova dimensão ao paradigma de navegação. Muitas áreas podem se beneficiar com este novo paradigma, dentre outras estão [Hoyos-Rivera, 2006]: ambientes de aprendizagem à distância, busca colaborativa de informações, navegação em materiais de suporte em vídeo-conferência.

CoLab (Collaborative Web Browsing) [Hoyos-Rivera, 2006] é uma ferramenta de co-navegação que fornece um modelo de sincronização simples e muito flexível permitindo, aos membros de sessões de co-navegação, criar e destruir dinamicamente relações de sincronização com outros membros da mesma sessão. Por tratar-se de uma ferramenta específica de co-navegação, CoLab não oferece mecanismos de comunicação entre os membros das sessões, o que impede que os mesmos venham a discutir acerca do conteúdo co-navegado. Portanto, torna-se necessário que CoLab seja utilizado em associação com ferramentas do tipo sistemas de mensagens ou ferramentas de áudio ou vídeo-conferência. Mas a simples utilização de CoLab em paralelo com uma ferramenta de comunicação exigiria que os próprios usuários coordenassem os dois sistemas manualmente. Um exemplo de comportamento seria garantir que os usuários co-navegando uma mesma página Web estivessem na mesma conferência.

A efetiva integração de CoLab a uma ferramenta de comunicação, como um serviço de conferência, deveria viabilizar uma coordenação automática dos dois sistemas. Desta forma, as operações realizadas em CoLab (por exemplo, operações de sincronização) poderiam conduzir à realização automática de operações no serviço de conferência, e vice-versa. Neste sentido, o presente artigo propõe a integração de CoLab com um Controlador de Conferências. Este controlador foi implementado usando-se o PBX IP Asterisk [Asterisk 2007] e oferece diversas funcionalidades de gerência de áudio e vídeo-conferências. Para a integração das duas ferramentas, foi adotado o ambiente LEICA (*Loosely-coupled Environment for Integrating Collaborative Applications*) [Gomes 2006]. LEICA permite uma integração fracamente acoplada de aplicações colaborativas possibilitando a abstração dos detalhes internos de cada aplicação, facilitando assim o processo de integração.

Este artigo está organizado na forma que se segue. A seção 2 aborda os trabalhos relacionados. As seções 3 e 4 apresentam os principais conceitos do ambiente LEICA e da ferramenta de co-navegação CoLab, respectivamente. Em seguida, a seção 5 descreve o Controlador de Conferências proposto neste artigo. Na seqüência, a seção 6 apresenta a integração do controlador proposto com a ferramenta de co-navegação CoLab. Finalmente, a seção 7 apresenta as conclusões e trabalhos futuros.

## **2. Trabalhos Relacionados**

A navegação Web colaborativa vem sendo investigada no contexto de diferentes projetos. Alguns trabalhos, como CoLab e CBW [Esenther 2002], propõem ferramentas específicas de co-navegação, sem suporte à comunicação. Outras ferramentas, incluindo CoBrowser [Maly 2001], IMMEX [Gerosa 2004] e PROOF [Cabri 1999], oferecem funcionalidades de comunicação baseadas em troca de mensagens. Além destes, existem ambientes completos de colaboração provendo mecanismos de co-navegação e suporte à comunicação com áudio e vídeo [Hong e Chen 2003] [Singh 2004]. Da mesma forma, alguns sistemas de VoIP (Voz sobre IP) [Wu e Schulzrinne 2004] [iVisit 2007] e Call Centers via Web [Yeo 2001] [Kim 2001] também suportam a co-navegação Web entre os membros de uma conferência.

É importante observar que a maioria dos sistemas citados suporta sessões de co-navegação apresentando as seguintes formas de organização: (i) não gerenciada, onde, a qualquer momento, qualquer usuário pode realizar uma ação de navegação; (ii)

centralizada, onde apenas um membro específico da sessão tem autonomia, ou o direito de navegar. Em ambos os casos, sempre que um usuário executar uma ação de navegação, os demais membros da sessão seguem passivamente a atividade de navegação deste primeiro. Como todos os usuários de uma sessão encontram-se sincronizados, a ferramenta de comunicação associada à sessão de co-navegação deve simplesmente manter esses usuários em uma mesma conferência.

O modelo de sincronização de CoLab permite que os usuários de uma mesma sessão criem de forma dinâmica relações de sincronização entre eles. Com isso, ao invés de impor uma organização centralizada (mestre-escravo) ou não gerenciada, este modelo é completamente flexível permitindo que, dinamicamente, a sessão possa ser organizada tanto de forma centralizada quanto descentralizada (onde usuários se distribuem em diferentes grupos de trabalho). Certamente, essa característica de CoLab torna o processo de integração a uma ferramenta de conferência mais complexo. A formação dinâmica de grupos de trabalho em uma mesma sessão de co-navegação exige a criação de diferentes conferências a serem associadas a esses grupos.

Uma limitação dos sistemas citados anteriormente é que a integração entre o mecanismo de co-navegação e as ferramentas de comunicação é realizada de forma *ad hoc*. Isto é, são utilizadas soluções de comunicação especificamente desenvolvidas para serem integradas aos respectivos sistemas. Além disso, torna-se inviável a substituição das ferramentas de comunicação oferecidas por outras disponíveis no mercado, comprometendo a flexibilidade desses sistemas. Visando uma solução genérica para o controle de conferências (independente de aplicação), Koskelainen (2002) define um *framework* baseado em componentes para a criação e controle de conferências. Protocolos padrões como SOAP (*Simple Object Access Protocol*) e SIP (*Session Initiation Protocol*) são empregados na comunicação com os servidores. Este *framework* foi implementado usando-se o servidor CINEMA e o cliente SIPc [Jiang 2001].

Neste trabalho é proposto um Controlador de Conferências que representa uma extensão do Asterisk. Ao contrário de Koskelainen (2002), a preocupação aqui não foi a definição de uma solução genérica, e sim, a criação de um sistema para o controle de conferências que possa ser facilmente integrado a outras ferramentas de colaboração. O Controlador de Conferências apresenta uma API que oferece a qualquer aplicação externa acesso direto às suas funcionalidades. Apoiando-se em LEICA, o Controlador de Conferências foi integrado a CoLab definindo-se assim um sistema de co-navegação integrado, onde as duas ferramentas são automaticamente coordenadas. É importante ressaltar que a integração através de LEICA favorece a flexibilidade e a extensibilidade do sistema, visto que outras ferramentas de colaboração ou comunicação podem igualmente ser integradas.

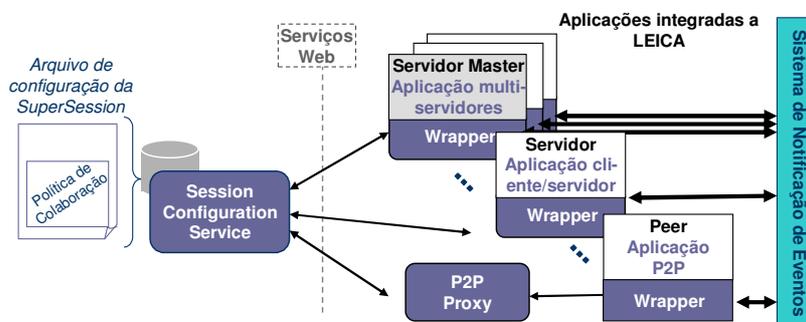
### **3. Ambiente de Integração LEICA**

Devido ao fato das atividades colaborativas envolverem diferentes indivíduos, apresentando diferentes requisitos e tarefas de grupo, o suporte a essas atividades é dificilmente oferecido por um sistema colaborativo simples. Conseqüentemente, os ambientes colaborativos atuais são constituídos por uma variedade de aplicações, trabalhando em paralelo, mas de forma independente. A integração destas aplicações pode trazer significantes benefícios para os usuários. De fato, um ambiente de colaboração

flexível deve permitir que diferentes funcionalidades de aplicações existentes sejam dinamicamente combinadas e coordenadas.

Com o objetivo de viabilizar a integração de diferentes aplicações colaborativas existentes, Gomes (2006) propôs LEICA – *Loosely-coupled Environment for Integrating Collaborative Applications*. Trata-se de um ambiente de integração que adota uma abordagem fracamente acoplada permitindo a abstração dos detalhes internos de cada aplicação, facilitando assim o processo de integração. 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. O grau de interação entre as aplicações integradas depende exclusivamente da API de cada aplicação, sendo que esta API caracteriza a natureza dos eventos que podem ser observados e as ações que cada aplicação é capaz de executar.

A Figura 1 ilustra o princípio arquitetural de LEICA. Para integrar as aplicações a LEICA, um *Wrapper* deve ser anexado aos servidores das aplicações colaborativas (ou aos *peers* de aplicações P2P). LEICA define um módulo específico chamado *API Factory*, que gera um *Wrapper* adaptado a cada aplicação baseando-se na descrição da respectiva API. Na verdade, dois arquivos de descrição devem ser criados: (i) Arquivo de Dados Específicos, que especifica quais são os dados necessários para criação de sessões convencionais para esta aplicação; (ii) Arquivo de Atributos e API, que especifica os atributos, os tipos de eventos que a aplicação é capaz de notificar, e os tipos de ações que ela que é capaz de executar através de sua API.



**Figura 1. Princípio arquitetural de LEICA**

Cada *Wrapper* compreende uma interface de serviços Web permitindo que aplicações colaborativas registrem-se em LEICA como uma aplicação integrada. Através dessa interface, as aplicações integradas podem igualmente interagir com o SCS (*Session Configuration Service*).

O SCS é um serviço Web usado para criar e iniciar novas *SuperSessions*. Uma *SuperSession* é uma sessão colaborativa integrada englobando todos os elementos de uma atividade colaborativa. Dentro do contexto de uma *SuperSession*, podem existir diferentes Sessões Específicas. Cada Sessão Específica é na verdade uma sessão colaborativa convencional definida por uma aplicação colaborativa (por exemplo, uma sessão de áudio-conferência ou de co-navegação web).

Durante a configuração de uma *SuperSession*, o SCS contata dinamicamente cada aplicação integrada para requisitar informações específicas sobre sua API. Estas informações serão usadas na criação de Sessões Específicas e especificação da Política de Colaboração de uma *SuperSession*.

Enquanto uma *SuperSession* é executada e as atividades colaborativas evoluem, as aplicações colaborativas envolvidas notificam a ocorrência de eventos entre elas (através do Sistema de Notificação de Eventos, baseado no paradigma *Publish/Subscribe*). Paralelamente a essas notificações, os *Wrappers* são encarregados de gerenciar a Política de Colaboração da *SuperSession*. Esta política é constituída de um conjunto de regras determinando como as aplicações devem reagir às notificações de evento. A Política de Colaboração permite, portanto, que seja definida para cada *SuperSession* uma semântica de integração específica (i.e. como coordenar as aplicações integradas) de acordo com diferentes requisitos dos usuários.

#### 4. Ferramenta de Co-navegação CoLab

CoLab [Hoyos-Rivera 2006] é um sistema de co-navegação que permite a organização dinâmica de uma sessão de co-navegação em diferentes grupos de trabalhos. Esta organização dinâmica é possível graças ao estabelecimento e a supressão de relações de sincronização entre membros de uma sessão. Estas operações são feitas via operadores de sincronização *I\_Follow\_You* e *You\_Follow\_Me* para criar relações de sincronização, e *I\_Leave\_You* para suprimir. CoLab adota o modelo WYSIWIMS (*What You See Is What I May See*), relaxando espacialmente a coerência dos dados de modo a permitir que cada utilizador observe secções diferentes da página web co-navegada.

Em CoLab, as relações de sincronização de cada grupo de trabalho são representadas por uma estrutura em árvore chamada SDT (*Synchronization Dependency Tree*). Em uma SDT, os nós representam os membros de um grupo de trabalho e as arestas representam as relações de sincronização entre eles. A Figura 2 ilustra dois SDTs que representam dois grupos de trabalho: um formado pelos usuários A, B e C; e outro pelos usuários D, E, F e G. Para um par de nós X e Y, se o nó X for pai do nó Y (aresta de X para Y), então a ação de navegação do usuário Y será sincronizada com aquelas do usuário X (Y “segue” a navegação de X). O nó raiz de cada SDT (nós A e D da Figura 2) representa um usuário assíncrono: o chefe do grupo de trabalho. Os demais membros do grupo, chamados síncronos, seguem as ações de navegação do respectivo chefe.

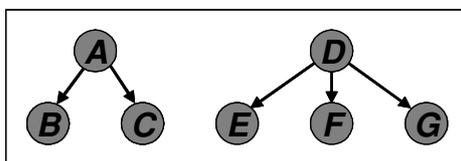


Figura 2. Cenários de configuração SDTs

A Figura 3 apresenta a arquitetura CoLab que segue uma abordagem baseada na utilização de um servidor Proxy. Quando um usuário assíncrono solicita uma ação de navegação, esta requisição é enviada ao servidor Proxy CoLab. O servidor satisfaz a requisição e informa a ocorrência desta ação aos navegadores membros síncronos do grupo, onde um agente (designado Controle de Navegação) repete a mesma ação.

O servidor CoLab possui cinco módulos:

- **Gerenciamento de Sessão** – Oferece funções de autenticação, autorização e de gerência da sessão. Uma de suas principais tarefas é tratar as ações de sincronização e garantir a coerência do estado global de sincronização.
- **API de Integração** – Fornece uma API que permite acrescentar novas funcionalidades ao CoLab, permitindo também integrá-lo a outras ferramentas de colaboração.
- **Proxy** – Intercepta as ações de navegação lançadas pelos usuários e pede ao Gerenciador de Sessão que verifique se as mesmas devem ser satisfeitas.
- **Gerenciamento de Navegação** – Responsável por tarefas ligadas à distribuição de recursos requisitados pelos usuários. Antes de apresentar as páginas Web, este módulo realiza uma modificação no código HTML a fim de permitir ao sistema interceptar as ações de navegação e sincronizar as apresentações de mídias contínuas.
- **Gerenciamento MediaSync** – Realiza as operações relacionadas à sincronização de apresentações de mídia contínua.

O cliente CoLab é composto de dois módulos:

- **Controle de Navegação** – Trata os comandos de apresentação da página Web corrente do usuário assíncrono (recebido do Proxy CoLab) e requisições para criar e suprimir as relações de sincronização recebidas dos usuários.
- **Controle de Mídia** – Notifica ao Gerenciador MediaSync qualquer mudança de estado das apresentações de mídia contínua (áudios e vídeos) em uma página Web e executa ações de controle de apresentação destas mídias solicitadas pelo Gerenciador de MediaSync.

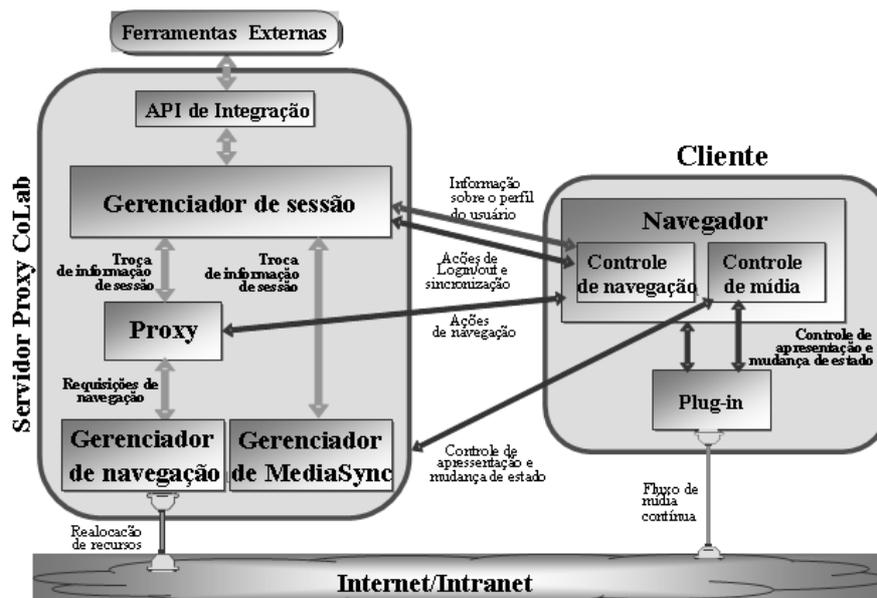


Figura 3. Arquitetura do CoLab

## 5. Controlador de Conferências para Ambientes Colaborativos

Segundo [Koskelainen 2002], um serviço de conferências deve permitir a criação, modificação e destruição de conferências, gerenciamento de usuários (adição e expulsão de participantes da conferência, e modificação de seus privilégios), e controle de palavra (ou *floor control*). [Koskelainen 2002] também aponta os requisitos gerais para um *framework* de controle de conferências genérico, sendo eles: ser suficientemente flexível para atender a maior parte dos cenários práticos de conferência; não ser muito complexo para ser aceito e implementado em dispositivos com recursos computacionais limitados; ser escalável (suportar grandes conferências distribuídas); ser de fácil extensão; genérico; confiável; e seguro. Além disso, para ser considerado genérico, um sistema de conferências deve suportar diferentes modelos de conferências, dentre os quais os mais relevantes são a conferência *dial-in* e *dial-out* [Rosenberg 2000]. Na conferência *dial-in*, os usuários devem conhecer o número da conferência e realizar manualmente uma chamada para participar desta conferência. Já na conferência *dial-out*, o servidor determina quais usuários farão parte da conferência e realiza uma chamada a cada um deles para que possam participar da mesma.

Um serviço de conferência pode ser de grande importância para um ambiente colaborativo, pois permite a comunicação direta entre seus participantes. Neste artigo, é proposto um serviço de conferência implementado na forma de um Controlador de Conferências. Este Controlador não possui o objetivo de representar uma solução genérica de serviço de conferência, muito menos um *framework* para o controle de conferências. Ele tem o objetivo de ser uma solução simples e específica de gerenciamento de conferências que possa ser facilmente integrada a outras aplicações colaborativas.

Como será detalhado na próxima sessão, o Controlador de Conferências define uma API flexível que facilita sua integração a outras aplicações. Em particular, apoiando-se no ambiente LEICA para implementar essa integração, não é necessário desenvolver uma solução proprietária de integração. Uma vez integrado a LEICA, o Controlador de Conferências poderá ser utilizado em conjunto com diferentes aplicações colaborativas integradas ao mesmo ambiente, viabilizando diferentes cenários de integração.

Outra característica do nosso Controlador de Conferência é o suporte apenas ao modelo de conferência *dial-out*, onde todo o gerenciamento de usuários e de conferências é diretamente realizado pelo Controlador em resposta a requisições recebidas das aplicações externas. Este controle automático simplifica o gerenciamento do trabalho colaborativo, visto que ele isenta os usuários de realizarem este controle manualmente (não precisando então decidir quando e em quais conferências eles devem se conectar).

Como descrito em [Siggelkow 2003], a divisão de tarefas entre diferentes grupos de trabalho durante realização de um trabalho colaborativo é uma prática comum visando à otimização deste trabalho. Torna-se, portanto, importante que um sistema de conferências ofereça mecanismos para suportar uma reorganização dinâmica de grupos de trabalho. Nesta direção, o Controlador de Conferências possibilita a divisão do grupo (representado pelos participantes de uma mesma sessão colaborativa) em diferentes conferências (formando grupos de trabalho). Para isto, foi introduzido no Controlador

de Conferências o conceito de sessão. Uma sessão é definida por um conjunto de membros (membros da sessão) e um conjunto de conferências. Cada conferência, por sua vez, é identificada por um identificador único e é associada a um subconjunto de membros da sessão.

### 5.1. Arquitetura do Controlador de Conferências

A Figura 4 apresenta os principais elementos que direta ou indiretamente interagem com o Controlador de Conferências proposto, sendo eles: (i) o servidor PBX IP Asterisk, (ii) o ambiente de integração LEICA, e (iii) os agentes usuários SIPs. Ele foi implementado em Java (sob a forma de uma aplicação standalone) com o objetivo de facilitar sua integração a LEICA, assim como a interação com o Asterisk.

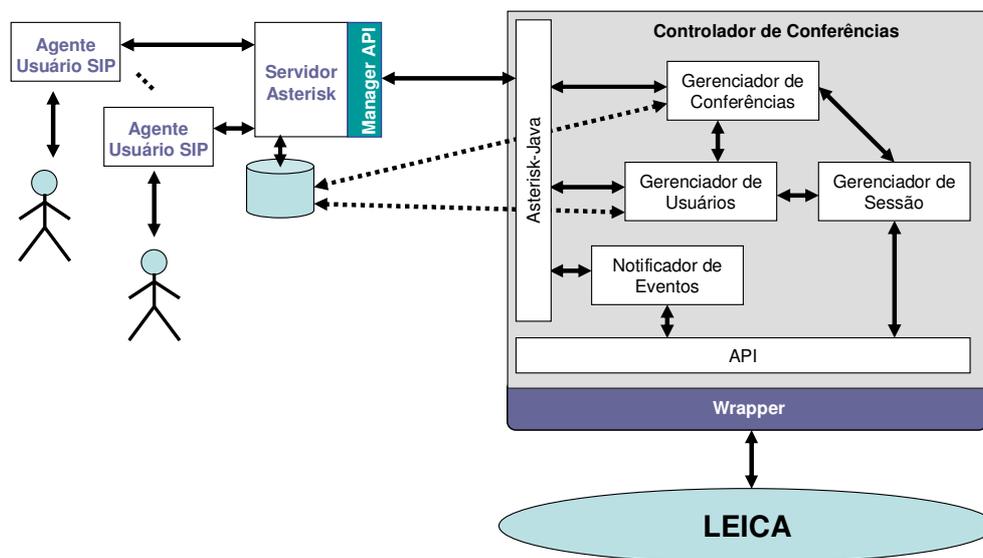


Figura 4. Arquitetura do Controlador de Conferências

As seções que seguem apresentam os principais componentes desta arquitetura.

### 5.2. Servidor Asterisk

O Controlador de Conferências é responsável por gerenciar conferências via o controle de um servidor PBX (*Private Branch Exchange*) IP Asterisk. Um PBX IP é um servidor que permite o estabelecimento de chamadas entre usuários VoIP, além de permitir que os seus usuários compartilhem certo número de linhas telefônicas convencionais. O Asterisk [Asterisk 2007] é um PBX IP largamente utilizado principalmente por ser de código aberto GPL (*General Public License*), além de oferecer diversas funcionalidades, como: serviços de email de voz, diretório, conferências, resposta interativa por voz (IVR), unidade de resposta audível (URA) e distribuidor automático de chamadas (DAC). Ele também suporta serviços de identificação de chamadas, ADSI, SIP, H.323 (como gateway e cliente) e MGCP (somente para gerenciamento).

O Asterisk tem como núcleo de seu sistema o plano de discagem, que define como devem ser tratadas as chamadas VoIP. Este plano consiste de uma lista de instruções ou passos personalizáveis que o Asterisk deve seguir de forma seqüencial no momento da ocorrência de cada chamada VoIP. Para o Asterisk, existem duas aplicações para o plano de discagem que dão suporte à criação de conferências:

app\_conference [App\_conference 2007] e Meetme [Mahler 2004]. A ferramenta app\_conference destaca-se por não necessitar de uma fonte de relógio externa para fornecer temporização nas conferências. Por outro lado, o Meetme possui mais recursos que o app\_conference, como administração de conferências, criação dinâmica de conferências e senha para ingresso em conferências. Por este motivo, a aplicação Meetme foi adotada neste trabalho.

O Controlador de Conferências pode controlar o funcionamento do servidor Asterisk via uma de suas API: Fast AGI (*Asterisk Gateway Interface*) [Meggelen 2005] ou a Manager API [Manager API 2007]. Usando a interface Fast AGI, as aplicações externas podem solicitar a execução de scripts AGIs contidas no plano de discagem do servidor Asterisk. No entanto, a aplicação externa não receberá nenhum resultado posterior destes scripts. Neste trabalho adotamos a Manager API, pois, além de solicitar a execução de funções no Asterisk, ela permite observar eventos ocorridos no interior do servidor Asterisk. Também foi utilizado o pacote Asterisk-Java [Asterisk-Java 2007] que define conjunto de classes Java que facilitam o uso tanto da Fast AGI quanto da Manager API.

### 5.3. Usuários e seus Agentes Usuários SIP

Enquanto o usuário interage com as demais aplicações colaborativas integradas a LEICA (no contexto de uma *SuperSession*), ele pode automaticamente receber chamadas para participar de conferências VoIP. Essas chamadas são, na verdade, solicitadas pelo Controlador de Conferências ao servidor Asterisk, que as envia então aos usuários. Cada chamada VoIP é finalmente tratada pelo agente usuário SIP (softphone, IP phone, *etc.*). Além de realizar convites aos usuários para participarem de conferências, o Controlador de Conferências pode requisitar o desligamento de um usuário de uma conferência. Da mesma forma, ele pode solicitar a criação e a destruição dinâmica de conferências.

É importante observar que uma vez integrado a LEICA, o Controlador de Conferências deverá realizar todas essas requisições ao servidor Asterisk em resposta ao ambiente de integração. A idéia é permitir que outra aplicação colaborativa integradas via o ambiente LEICA controlem o sistema de conferências em função da atividade colaborativa suportada por essas aplicações.

### 5.4. Controlador de Conferências

Como se encontra detalhado na Figura 4, as funcionalidades do Controlador de Conferências são implementadas por diferentes componentes:

- **Wrapper LEICA e a API do Controlador de Conferências** – Seguindo o processo de integração descrito na seção 3, o *Wrapper* foi gerado e associado ao Controlador de Conferências através de sua API.
- **Notificador de Eventos** – É responsável por repassar alguns eventos ocorridos no servidor Asterisk para as aplicações externas, através da API do Controlador (por exemplo, a entrada e saída de participantes da conferência).
- **Gerenciador de Sessão** – Implementa o conceito de sessão, efetuando todas as operações de criação e destruição de sessão. Este componente mantém todos os registros das sessões criadas (incluindo as conferências criadas no contexto de

uma sessão e os membros que participam de cada conferência). No momento da criação de uma sessão deve-se informar a lista de usuário que serão membros desta sessão. Neste momento, este módulo interage com o Gerenciador de Usuários para a validação destes usuários (é verificado se os mesmos encontram-se cadastrados na base de dados do servidor Asterisk).

- **Gerenciador de Usuários** – Mantém informações sobre os participantes das sessões e conferências (com base em consultas e alterações na base de dados do Asterisk). Este gerenciador também tem acesso a base de dados do servidor Asterisk, permitindo obter diversas informações, como o *codec* utilizado pelo participante, email de voz, e endereço IP atual.
- **Gerenciador de Conferências** – É responsável por criar e destruir conferências, assim como incluir e excluir participantes das conferências. Na operação de criação de conferência, é informado o identificador da sessão na qual a conferência deve ser criada e a lista de participantes. Para executar essa operação, este componente interage com o Gerenciador de Sessão para registrar a conferência. Em seguida, o gerenciador aloca um número para a mesma. Finalmente, o Gerenciador de Conferências solicita a criação da conferência junto ao servidor Asterisk, solicitando igualmente as chamadas VoIP para os participantes desta conferência. Na operação de destruição de conferência, este componente também deve interagir tanto com o Gerenciador de Sessão quanto com o Asterisk para realizar o encerramento efetivo de uma conferência SIP.

## 6. Co-navegação com Suporte a Conferência

Para que uma sessão de co-navegação possa ser efetivada, é necessária uma ferramenta de comunicação onde os membros da sessão possam discutir sobre o conteúdo que está sendo co-navegado. Desta forma, CoLab pode ser integrado ao Controlador de Conferências, fazendo uso de LEICA como ambiente de integração. Vale observar que uma vez integrados a LEICA, CoLab e o Controlador de Conferências podem ser utilizados em diferentes cenários de integração, associados a outras aplicações colaborativas integradas a LEICA.

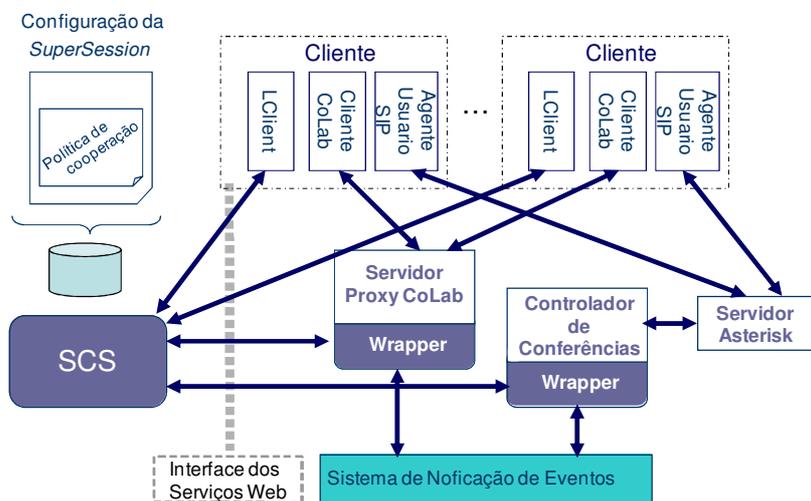
Para facilitar a definição de uma *SuperSession* global, é importante definir o conceito de sessão para cada sistema. Cada sessão CoLab deve, então, ser associada a uma sessão do Controlador de Conferências. No contexto de uma sessão CoLab, diferentes grupos podem ser formados durante a colaboração, onde cada grupo deve ser associado a uma áudio-conferência. Desta forma, enquanto os grupos são dinamicamente criados e destruídos em uma sessão CoLab (como resultado da operação de sincronização dos usuários), as respectivas conferências precisam ser criadas/destruídas pelo Gerenciador de Conferências.

A Figura 5 apresenta os principais componentes do ambiente de co-navegação:

- **Serviço de Configuração de Sessão (SCS)** – O SCS de LEICA é usado para criar e iniciar *SuperSessions*. Criamos o arquivo de configuração da *SuperSession* que chamamos de “CoLabMaisConferência” de acordo com o formato especificado por LEICA. Este arquivo especifica 3 tipos de informação: (i) informação de gerência geral (nome e descrição da sessão, informações dos membros, *etc.*); (ii) informação de configuração específica para o CoLab e o Controlador de Conferências; (iii) uma Política de Colaboração (um conjunto de

regras para coordenar a co-navegação sistema de conferências). É importante observar que é a especificação desta Política de Colaboração que permite implementar o comportamento descrito anteriormente, onde cada grupo de trabalho de CoLab deve ser associado a uma áudio-conferência específica.

- **Servidor Proxy Colab e seus Wrappers** – O processo de integração descrito na sessão 3 foi executado para gerar automaticamente um *Wrapper* adaptado à API de integração CoLab. Posteriormente, este *Wrapper* é integrado ao servidor Proxy CoLab utilizando-se justamente essa API.
- **LEICA Client (LClient)** – A fim de se ingressar em uma *SuperSession*, o usuário precisa executar a aplicação *LClient*. O *LClient* inicialmente se conecta ao SCS e recebe as informações de configuração para fornecer a *SuperSession* (escolhida pelo usuário). Baseando-se na Política de Colaboração, *LClient* inicia o cliente CoLab (carregado em um navegador web contendo a pagina inicial da sessão de co-navegação). Quanto ao agente SIP, este deve ser previamente executado pelo usuário, permitindo assim que o mesmo receba chamadas VoIP. Desta forma, o usuário tem a liberdade de utilizar qualquer agente usuário SIP.
- **Sistema de Notificação de Eventos** – Baseado neste sistema, tanto CoLab como o Controlador de Conferências notificam eventos durante a *SuperSession* (por exemplo, indicando que o usuário A sincronizou com usuário B na sessão de co-navegação). Paralelamente, a Política de Colaboração é continuamente analisada pelos *Wrappers* a fim de verificar se uma ação precisa ser executada em resposta a um evento notificado (por exemplo, o Controlador de Conferências precisa criar uma áudio-conferência conectando o usuário A e B quando estes se juntarem ao mesmo grupo de trabalho CoLab).

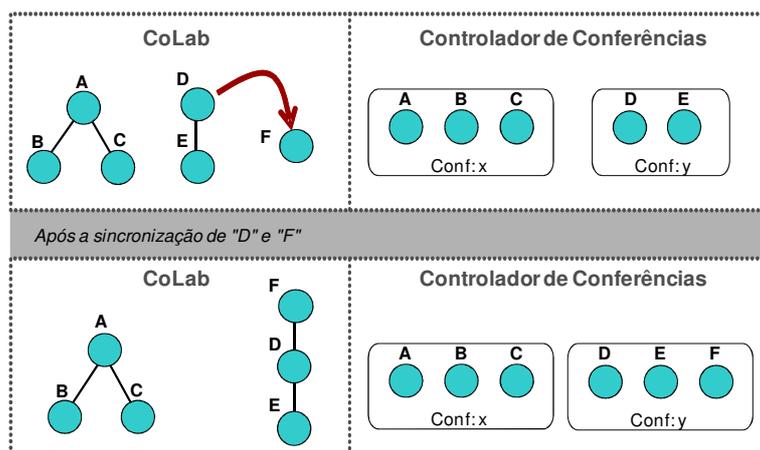


**Figura 5. Ambiente de Co-navegação**

A fim de ilustrar o uso do nosso ambiente de co-navegação, considere seis usuários participando de uma sessão de co-navegação (A, B, C, D, E, e F). Suponha também que para essa sessão foi definida a seguinte semântica de integração: cada grupo de trabalho do CoLab é associado a uma conferência, permitindo a usuários de um mesmo grupo discutirem sobre sua atividade de navegação utilizando um áudio-conferência específica.

Quando os usuários se conectam ao CoLab, eles iniciam suas atividades de navegação de forma independente (representando usuários assíncronos). Durante a sessão, um usuário pode decidir perder sua autonomia de navegação e começar a “seguir” as atividades de navegação de outro participante. Neste instante, o Controlador de Conferências comanda uma chamada VoIP para esses dois usuários a fim de que eles participem de uma conferência. Caso o segundo usuário já faça parte de um grupo de trabalho, apenas o primeiro (o que acaba de se tornar síncrono) receberá a chamada.

Na medida em que as relações de sincronização são criadas ou liberadas entre os usuários, serão estabelecidas diferentes SDTs representando os diversos grupos de trabalho. A parte superior da Figura 6 ilustra o instante em que os usuários A, B e C se encontram no mesmo grupo de trabalho (uma SDT), e D e E formam um segundo grupo de trabalho, enquanto F encontra-se assíncrono. Observe que o Controlador de Conferências criou duas conferências: a conferência “x” na qual participam os usuários A, B e C; e a conferência “y” com D e E. Se D decide sincronizar com F, este último se torna raiz da SDT inicialmente formada por D e E. Neste momento, o Controlador de Conferências é informado sobre a operação de sincronização (através de uma notificação de evento enviada pelo CoLab), e em resposta convida o usuário F para a conferência “y” conectando B e D.



**Figura 6. Cenário de Co-navegação**

A semântica de integração adotada neste exemplo (onde cada grupo de trabalho de CoLab é associado a uma áudio-conferência específica) é definida através da Política de Colaboração da *SuperSession*. Para que outros tipos de comportamento sejam especificados, basta alterar as regras dessa Política. Por exemplo, (i) pode-se manter um usuário em uma áudio-conferência mesmo que ele se desligue de um grupo de trabalho CoLab (enquanto ele não se juntar a um novo grupo); ou ainda (ii) pode-se definir que todos os membros da sessão de co-navegação sejam incluídos em uma única áudio-conferência.

## 7. Conclusões e Trabalhos Futuros

Neste artigo, apresentamos um ambiente de co-navegação baseado na integração da ferramenta de co-navegação CoLab e o Controlador de Conferências. Esta integração foi realizada graças ao sistema LEICA. A adoção de LEICA simplificou significativamente

a integração destas duas aplicações, e poderá igualmente facilitar a integração de novas aplicações ao nosso ambiente.

A integração de CoLab com o Controlador de Conferências permite que operações de sincronização dos membros das sessões CoLab passem a controlar a entrada e saída destes membros em diferentes conferências. Esta operação automática evita que os usuários tenham que manualmente controlar qual conferência eles devem chamar para discutir o assunto co-navegado, facilitando, portanto a gerência da sessão.

A possibilidade de se definir diferentes políticas de colaboração no ambiente LEICA permite que o sistema proposto seja flexível e adaptável a diferentes cenários de colaboração: do caso mais simples, da inclusão de todos os membros da sessão em uma única conferência; a cenários mais complexos como manter membros de grupos de trabalhos dinâmicos em conferências distintas.

Quanto aos trabalhos futuros, será investigada a inclusão de mecanismos de controle de palavra no Controlador de Conferências. Uma vez estes mecanismos implementados, uma semântica de integração mais elaborada poderá ser definida para a *SuperSession*, criando-se novas regras na sua Política de Colaboração. O objetivo será associar privilégios ou permissões especiais a tipos diferentes de usuários (LEICA suporta a associação de papéis ou *roles* aos usuários de uma *SuperSession*). Por exemplo, permitir que apenas usuários de tipo “Professor” possam controlar a palavra em uma áudio-conferência.

Também será importante verificar e melhorar a escalabilidade do Controlador de Conferências para que o mesmo seja capaz de suportar grandes conferências distribuídas. Uma solução possível é a implementação de conferências descentralizadas utilizando-se diferentes PBXs IP Asterisk. O problema da escalabilidade também deverá ser investigado no caso de CoLab, empregando-se, por exemplo, diferentes Proxies.

A fim de validar o sistema proposto, este será utilizado em um cenário de *e-learning* no contexto do projeto SIDIE [SIDIE 2007].

## **Agradecimentos**

O presente trabalho foi realizado com o apoio do CNPq (Proc. N° 506789/04-7).

## **Referências**

- Asterisk. (2007) “The Open Source PBX IP”. <http://www.asterisk.org/>, Março.
- Asterisk-Java. (2007) ”Java control for Asterisk PBXes”. <http://asterisk-java.org>, Dezembro.
- App\_conference. (2006) Asterisk app\_conference. [http://www.voip-info.org/wiki-Asterisk+app\\_conference](http://www.voip-info.org/wiki-Asterisk+app_conference), Março.
- Cabri, G., Leonardi, L., Zambonelli, F. (1999) “A Proxy-based Framework to Support Synchronous Cooperation on the Web. Software - Practice and Experience”, 29(14), p. 1241–1263.
- Yeo, C.K., Hui, S.C., Soon, I.Y. (2001) “A Multimedia Call Centre on the Internet”. Int. Journal of the Computer, The Internet and Management, Vol. 9, No.1, pp. 1-12.

- Kim, D.H. Park, S.M., Kim, J.Y., Sul, D.M., Lee, K.H. (2001) "Collaborative Multimedia Middleware Architecture and Advanced Internet Call Center". ICOIN, pp. 246-252.
- Esenther A.W. (2002) "Instant Co-Browsing: Lightweight Real-time Collaborative Web Browsing". 11th International WWW Conference, pp. 330-338.
- Gerosa, L., Giordani, A., Ronchetti, M., Soller, A., Stevens, R. (2004) "Symmetric Synchronous Collaborative Navigation". IADIS International Conference WWW/Internet: p. 748-754.
- Gomes, R.L., Hoyos-Rivera, G.J., Courtiat, J.P. (2006) "Um Ambiente para Integração de Aplicações Colaborativas". Simpósio Brasileiro de Sistemas Colaborativos.
- Hong, H. C., Chen, Y. C. (2003) "Design and Implementation of a Web-Based Real-Time Interactive Collaboration Environment". FTDCS p. 295-300.
- Hoyos-Rivera, G.J.H., Gomes, R.L., Courtiat, J.P., Willrich, R. (2006) "CoLab A New Paradigm and Tool for Browsing Collaboratively the Web". IEEE Trans. on Systems, Man and Cybernetics. Part A, Systems and Humans, v.36, n.6, pp. 1074-1085.
- iVisit (2007). <http://www.ivisit.com>, Março.
- Jiang, W., Lennox, J., Schulzrinne, H., and Singh, K. (2001) "Towards junking the PBX: deploying IP telephony". In Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV).
- Koskelainen P., Schulzrinne H., Wu X. (2002) "A SIP-based conference control framework", NOSSDAV, p. 53-61.
- Mahler, P. (2004) "VoIP Telephony with Asterisk". Ed. Signate, 2 Edição.
- Maly, K., Zubair, M., Li, L. (2001) "CoBrowser: Surfing the Web Using a Standard Browser". World Conference on Educational Multimedia, Hypermedia and Telecommunications, p. 1220-1225.
- Manager API (2007) Asterisk Manager API. <http://www.voip-info.org/wiki-Asterisk+manager+API>, Março.
- Siggelkow, N., Levinthal, D. (2003) "Temporarily divide to conquer: Centralized, decentralized," *Org. Sci.*, vol. 14, no. 6, p. 650-669.
- SIDIE "Sistema de Disponibilização de Informações para o Ensino" (2007). <http://www.sidie.inf.ufsc.br>, Abril.
- Rosenberg J., Schulzrinne H. (2000) "Models for multiparty conferencing in SIP," Internet Draft, Internet Engineering Task Force.
- Singh, K, Wu, X., Lennox, J., and Schulzrinne, H. (2004) "Comprehensive Multi-platform Collaboration", MMCN 2004, SPIE Conf. on Multimedia Computing and Networking.
- Wu, X., Schulzrinne, H. (2004) "SIPC, a Multi-function SIP User Agent". MMNS 2004, LNCS 3271, p. 269-281.

# Modelo de Design Rationale para atendimento médico ambulatorial

Cleo Billa<sup>1</sup> Cláudia Barsottini<sup>2</sup>, Jacques Wainer<sup>1,2</sup>

<sup>1</sup>Instituto de Computação - Universidade Estadual de Campinas (Unicamp)

<sup>2</sup>Departamento de Informática em Saúde – Universidade Federal de São Paulo (Unifesp)

{cleo,wainer}@ic.unicamp.br, claudia@dis.epm.br

**Abstract.** *In a health care environment, it is not very usual interactions between of health care professionals. This work shows that electronic medical records available do not favor this kind of interaction and propose a design rationale based model applied to health care. This model is an extension to the electronic medical record and it intends to help health care professionals on collaborative work.*

**Resumo.** *Num ambiente de trabalho em saúde, não é muito comum que haja interação entre os profissionais de saúde. Esse trabalho mostra que os sistemas de prontuário eletrônico disponíveis atualmente não favorecem esse tipo de interação e propõe um modelo baseado em design rationale aplicado à saúde. Esse modelo é uma extensão ao prontuário eletrônico e tem como principal finalidade ajudar os profissionais de saúde no trabalho colaborativo.*

Key-words: collaborative work practices, integration, patient record, design rationale

## 1. Introdução

Pesquisas na área de tecnologia de informação integrando o trabalho em saúde com sistemas colaborativos eficientes têm sido cada vez mais desafiantes. Isto é, entender como esses profissionais de saúde registram e utilizam as informações coletadas do paciente durante o atendimento do mesmo [Hartswood et al., 2003; Alberdi, et al., 2005; Clarke et al., 2006]. Este entendimento busca oferecer ao trabalho do profissional de saúde uma maior qualidade no cuidado tornando-o mais eficiente e mais centrado no paciente a partir da introdução de novas tecnologias [Berg 99; Berg99].

Uma das características do cuidado em saúde é o fato de que um paciente pode ser atendido por uma equipe de saúde multiprofissional composta por: médicos, enfermeiros, nutricionistas, fisioterapeutas, entre outros, sendo assim, a informação coletada por um profissional pode servir para outro profissional da equipe. Por outro lado, isso permite que

cada grupo de profissionais de saúde tenha seu próprio padrão de atendimento dentro da sua especialidade. Entretanto, a divisão física de setores, equipes, especialidades e hierarquias descentralizadas dentro dos serviços de saúde, dificultam muito a permeabilidade das informações do paciente, pelos profissionais que prestam o serviço [Pinelli 2006].

A partir da introdução do prontuário eletrônico nas organizações de saúde, acreditou-se que haveria uma maior fluidez das informações propiciando a colaboração entre os profissionais de saúde em particular os médicos, porém algumas pesquisas demonstram limitações do prontuário eletrônico quando são usados por equipes multidisciplinares [Ellis 1991, Strauss 1985]. Esta uma das razões pela qual o prontuário do paciente muitas vezes se mostra ineficiente em promover a colaboração entre os profissionais de saúde. Outro motivo descrito na literatura que justifica a falta de colaboração entre médicos, é que o que está escrito no prontuário do paciente é substancialmente distinto da prática médica, não importando se o registro é feito em papel ou eletronicamente [Berg 1999]. Geralmente, há uma falta de integralidade do registro e de mecanismos de consistência da entrada da informação induzindo o médico agir de modo omissivo no atendimento do paciente, fornecendo apenas um esboço racional para justificar decisões administrativas e essa omissão é importante não somente para a atribuição de responsabilidades, mas apoiada em seu modo visual, vai guiando e induzindo as ações dos outros, na atividade de “dar uma olhada”[Clark 2002]. De forma pragmática, o prontuário do paciente reproduz informações redundantes, fragmentadas e ambíguas sobre o paciente [Atikson 1995].

Mais recentemente, algumas pesquisas em sistemas de informação em saúde apontam desafios para que o prontuário do paciente aprimore a colaboração entre os profissionais da saúde e proporcione melhorias diretas ao cuidado do paciente [Hartwood 2003, Xiao 2005]. Aplicações distribuídas e com enfoque colaborativo nos registros eletrônicos dos pacientes tem sido apontadas como exigências chaves para serviços de saúde mais eficazes, de melhor qualidade e de custos mais baixos [Pinelle 2006]. No entanto, não parece ser claro que as circunstâncias atuais da prática em saúde possam fornecer uma boa base para que sistemas colaborativos em saúde sejam bem sucedidos. Por outro lado, não se pode negar o fato que muitos dos sistemas de informação em saúde falharam pela dificuldade sistemática em esclarecer a natureza cheia de exceções do trabalho médico [Pratt 2004].

Em um ambiente de trabalho colaborativo é importante para a interação e comunicação humana em geral, compreender as atividades dos outros, assim como a consciência e a responsabilidade também são fatores importantes, pois tornam os ambientes colaborativos dinâmicos e confiáveis, principalmente na área da saúde. Estes são fatos importantes que devem ser levados em consideração no desenvolvimento de arquiteturas distribuídas e baseadas em CSCW (Computer System Collaborative Work). [Weerakkody2003].

O alvo deste estudo é identificar as exigências sociais e organizacionais de um grupo de profissionais médicos, focado no uso de sistema informatizado para registro das informações do paciente, e desenhar uma interface com base em “Design Rationale” que proporcione colaboração entre eles.

## **2. Relevância da pesquisa em colaboração médica**

Esse estudo tem como um dos seus principais objetivos entender qual a natureza da interação entre médicos que assistem a um mesmo paciente por longos períodos de tempo, tendo como meio de comunicação o prontuário e identificar como extraem e utilizam a informação registrada no prontuário para a elaboração do cuidado em saúde. Pode-se entender como sendo uma colaboração através do tempo: um primeiro médico que atendeu um paciente por algum tempo, registrou informações sobre ele no prontuário, e um segundo médico, que ao atender esse paciente resgata essas informações para entender qual a histórico da doença do paciente e dar continuidade ao tratamento. Este estudo descreve que as informações normalmente disponíveis em prontuários não são suficientes para que essa colaboração através do tempo seja efetiva.

Foram estudados dois ambulatórios clínicos em um grande hospital universitário da cidade de São Paulo. O hospital deste estudo tem por característica ser público e de alta complexidade, onde o paciente pode dar entrada para uma variedade enorme de procedimentos; indo desde um pronto atendimento, consultas periódicas, ambulatórios especializados, exames ou até uma internação. Os serviços deste hospital são custeados pelo governo, e assim o hospital atrai principalmente a população sem assistência médica privada. O detalhamento e características dos setores estudados esta descrito no trabalho de Barsottini et. al de 2005. [Barsottini et. al, 2005]

A prática em saúde, em especial o tratamento a longo prazo dos pacientes com doenças crônicas ou síndrômicas, parece ser um exemplo particularmente interessante de trabalho que deva promover a colaboração entre médicos. No exemplo de doenças síndrômicas, os pacientes apresentam um conjunto de sinais e sintomas e um único médico pode não ser capaz de cuidar de todos os aspectos da síndrome, outro aspecto destas doenças que precisa ser levado em consideração é o tempo da doença, que por definição podem perdurar por toda uma vida.

Em um ambiente acadêmico a rotatividade dos profissionais é alta, dado o caráter de formação profissional, ou seja, em um serviço que atende pacientes crônicos, o paciente é atendido por vários profissionais ao longo de sua vida. Mediante este panorama, a situação de colaboração entre especialistas torna-se desejável. Já nos casos de doenças agudas, a informação ambígua ou faltosa sobre o paciente pode comprometer em muito o diagnóstico rápido e preciso. Também nesta situação, a colaboração entre médicos mostra-se necessária. Por outro lado, uma quantidade não gerenciável de informações dificulta a organização dos membros do grupo, ocasionando desentendimentos [Fussel et al, 1998]. Vale ressaltar, que a existência da sobrecarga de informação está extremamente ligada ao indivíduo. Alguns conseguem lidar com mais informações simultâneas do que outros, dependendo, entre outros fatores, da maturidade, das capacidades e das habilidades de cada um, bem como das características e do nível de conhecimento sobre o assunto em questão. Portanto é necessário o controle para que o fluxo de informações não seja maior do que a capacidade do indivíduo de processá-la e digeri-la, apesar desta capacidade não ser facilmente mensurável. Para evitar a sobrecarga, é necessário balancear a necessidade de

fornecer informações com a de preservar a atenção sobre o trabalho. O fornecimento de informações de forma assíncrona, estruturada, filtrada, agrupada, resumida e personalizada facilita esta tarefa [Kraut and Attewell, 1997]. Assim, fornece-se uma visão geral para que o médico selecione em que parte da informação deseja trabalhar, e o detalhamento é obtido quando for demandado. A redução da sobrecarga de informação na comunicação, por exemplo, pode se dar através do armazenamento das justificativas de uma tomada de decisão de um tratamento e do fornecimento de informações simples e representativas que ajudem os médicos a identificar a relevância e o contexto das decisões, sem que o prontuário seja lido completamente [Gerosa et al, 2001].

### **3. Estudo de caso**

Este estudo segue a linha de utilização de métodos qualitativos para levantamento, análise de dados e descrição dos artefatos encontrados, com a clara proposta de responder às perguntas da pesquisa e dar suporte para o desenvolvimento de uma interface colaborativa baseada em Design Rationale. As observações de campo e as entrevistas com os médicos foram usadas conseqüentemente para o levantamento de dados. O método de observação não participativa e entrevistas não estruturadas tiveram como objetivo reafirmar as percepções encontradas pelos autores ao descrever a atuação dos médicos, usando o prontuário para atender aos seus pacientes, e situações crônicas de difícil colaboração entre médicos pelo uso do prontuário [MacLean, 2006]. Dois ambulatórios de doenças crônicas participaram do estudo, o Ambulatório de Neurologia Geral e o de Litíase Renal.

A neurologia clínica é uma especialidade onde a maioria dos casos é crônica e sindrômica, em que o paciente é controlado em consultas periódicas, que avaliam a evolução da doença, os ajustes nos medicamentos, o controle evolutivo de alguns sintomas e a necessidade de interconsultas com outras especialidades. O atendimento ao paciente requer discussões extensivas, coletivas e interdisciplinares entre os profissionais de saúde para a melhor tomada de decisão. Muito das informações dos pacientes ficam retidas junto ao médico chefe, que vê todos os casos continuamente.

A especialidade de Nefrologia, mais particularmente a subespecialidade de litíase renal, tem por característica geral atender pacientes crônicos em consultas periódicas que avaliam a evolução da doença e a necessidade de ajuste nos medicamentos. Assim como no caso anterior o médico chefe discute todos os casos e acaba por reter grande parte das informações dos pacientes atendidos.

Em ambos os ambulatórios os prontuários são extensos devido ao tempo de doença do paciente, além de serem incompletos pois o paciente tem outros prontuários em outras especialidades. Vale lembrar que por características do serviço de saúde e do hospital ser universitário, esse paciente nem sempre é atendido por um mesmo médico. O atendimento ao paciente requer discussões extensivas, coletivas e interdisciplinares entre os profissionais de saúde para a melhor tomada de decisão. Muito da informação do paciente fica retido junto ao médico chefe, que discute todos os casos continuamente com residentes

e com profissionais de outros setores do hospital.

Por meio desta pesquisa [Barsottini,2005] fomos levados a descobertas de fatos que podem estar contribuindo para a não colaboração entre os médicos por meio do prontuário, como:

- Linguagem inadequada
- Falta de informação sobre a hipótese diagnóstica.
- Falta de informação sobre terapêutica usada e medicações (posologia, duração, início e término, relação com os sintomas).
- Inadequação da forma de coleta e apresentação dos dados.
- Falta de relato sobre a discussão com preceptores sobre o caso do paciente.

Os achados encontrados neste estudo de caso nos levaram a prosseguir em busca dos motivos e da quantidade da falta de colaboração entre médicos no uso do prontuário eletrônico.

#### **4. Proposta do modelo**

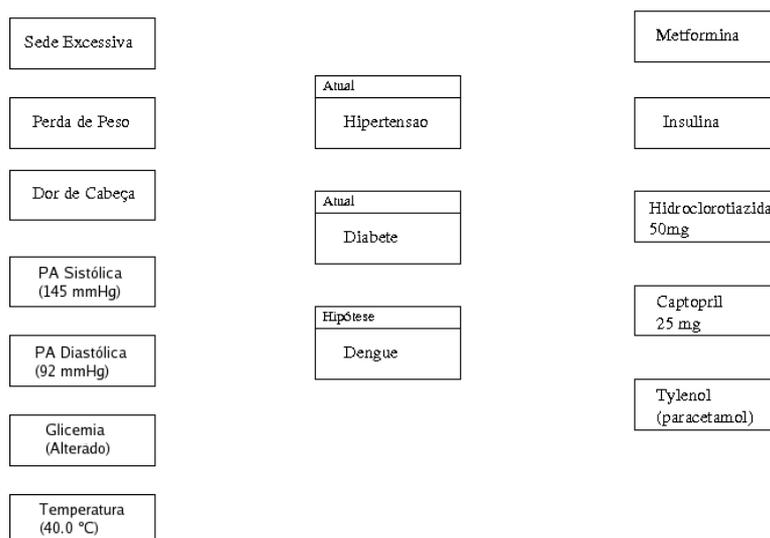
Nesse trabalho propõe-se a utilização de um modelo interface baseada em design rationale como um adendo no prontuário eletrônico. Esta interface tem como objetivo possibilitar a colaboração entre médicos. Inicialmente pretende-se além de analisar as informações colaborativas entre dois médicos, estudar os efeitos desta colaboração no tratamento do paciente e na satisfação do profissional de saúde. Em nossa pesquisa sobre a pouca colaboração entre médicos, encontramos alguns problemas que são desafios que se espera que a interface proposta deva suprir, apresentando uma conexão do histórico das várias consultas do paciente, reduzindo a falta de informações e modificando a coleta e apresentação de alguns dados. O modelo é baseado numa representação gráfica onde o profissional de saúde pode relacionar os achados clínicos com decisões e ações tomadas, assim como justificá-las.

Neste modelo gráfico os sinais, sintomas, exames, diagnósticos e medicamentos são representados por retângulos, as relações entre eles são representados por setas e os comentários feitos pelos profissionais por elipses. Todos os objetos do modelo podem ser relacionados entre si.

A base deste modelo é o prontuário eletrônico do paciente, onde por definição o profissional de saúde deve inserir todo tipo de informação que achar necessária para relatar o atendimento dado ao paciente. O modelo de interface proposto, constrói a partir destas informações uma representação gráfica das informações da consulta. Por exemplo, um paciente chega a uma consulta com as seguintes queixas: “dor de cabeça que pressiona a nuca”, “que nos últimos dias está sempre com sede” e “que tem notado que perdeu peso nos últimos 5 dias”. Durante o exame físico geral, o médico mede a pressão arterial do paciente, a temperatura, e faz um exame de glicemia capilar. As informações colhidas pelo médico durante o atendimento levam a hipótese diagnóstica de: pressão elevada, glicemia elevada e um quadro gripal que pode ser também uma suspeita de dengue. Ao final da consulta o

médico conversa com o paciente sobre as suas hipóteses de diagnóstico, prescreve medicamentos, faz orientações gerais, agenda o retorno e a consulta é encerrada.

O médico insere no prontuário as informações relacionadas ao paciente, história do paciente, sintomas, exames, diagnóstico e tratamento. Como comentado anteriormente não existe nenhum tipo de controle sobre a forma ou conteúdo da informação inserida principalmente no que diz respeito à justificativa da decisão tomada. Desta forma, outro profissional de saúde que ler este prontuário pode se perguntar “porque foi dado esse medicamento?”, ou “o que levou a um determinado diagnóstico?”. Para poder responder a esse tipo de pergunta ou simplesmente compreender melhor as ações e decisões do profissional de saúde, o modelo aqui proposto contém as principais informações da consulta permitindo que elas sejam relacionadas e/ou justificadas. A figura 1 mostra a representação gráfica do modelo gerada automaticamente pelo sistema. Essa representação gráfica contém todos os sintomas, exames, diagnósticos e medicamentos do paciente. A partir desse “layout” o profissional de saúde deve expressar o seu raciocínio gerando relações e justificativas entre as suas ações. Ou seja, no gráfico gerado ele deve relacionar as informações e se desejar inserir comentários sobre as ações e decisões tomadas. Para expressar as relações basta que ele puxe setas indicando que duas informações estão diretamente relacionadas. Por exemplo, na figura 1 ele pode mostrar a relação existente entre o exame de glicemia e o diagnóstico de diabetes. Ou então, ele pode relacionar o medicamento *captopril* com hipertensão. Como a idéia é que o profissional de saúde expresse o seu raciocínio, essas relações não têm nenhum tipo de restrição. O profissional tem total de liberdade de criar relações. Ele pode inclusive fazer mais de uma associação para uma mesma informação, ou seja, o medicamento *hidroclorotiazida* também é indicado para quem tem hipertensão, portanto dois ou mais medicamentos comumente podem estar associados a um mesmo diagnóstico.



**Figura 01 - Gráfico gerado pelo sistema durante a consulta**

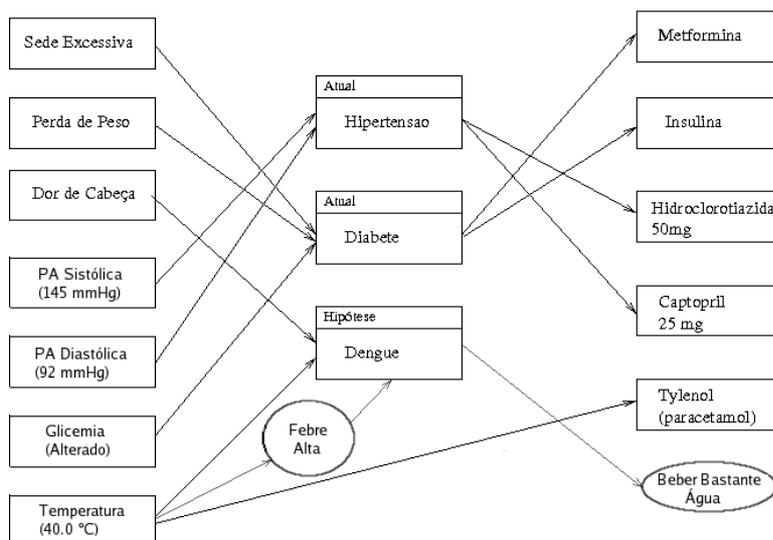
Um exemplo de como as relações podem ser feitas é mostrada na figura 2. Pode-se observar pela figura, que os sintomas sede excessiva e perda de peso em conjunto com o exame de glicemia levaram ao diagnóstico de diabetes. Assim como a medição da pressão arterial levou ao diagnóstico de hipertensão. Os outros sintomas como dor de cabeça e febre alta levaram a suspeita de dengue. Também pode ser visto pela figura 2 o que cada medicamento procura tratar, inclusive pode ser observada uma relação direta entre um medicamento e um exame (*tylenol* para febre alta).

O objetivo do modelo é tentar explicitar como o profissional de saúde chegou a um determinado diagnóstico, ou com que intenção foi prescrito tal medicação, ou ainda porque foi pedido tal exame. Assim, quanto mais descritivo for esse modelo, mais fácil será o entendimento do raciocínio do profissional durante a consulta.

Como já foi dito, o modelo também permite inserir comentários e relacionar esses com qualquer outra informação do gráfico. Na figura 2 o profissional relaciona o exame de temperatura do paciente com o comentário “Febre Alta” e ainda relaciona esse com o diagnóstico de dengue. Expressando assim seu raciocínio para o diagnóstico. Ele ainda faz outro tipo de comentário: “Beber bastante água”, mas nesse caso ele só informa o que foi sugerido ao paciente para melhorar o tratamento. Dessa forma, pode-se observar que a utilização de comentários pode servir com qualquer propósito, seja para a justificativa de um diagnóstico, ou para uma simples informação complementar para um tratamento.

Com a possibilidade de inserção de comentários no gráfico da consulta, o sistema permite que o profissional de saúde expresse o seu raciocínio de forma mais detalhada. Quando outro profissional de saúde quiser analisar o que foi feito nas consultas anteriores, ele vai encontrar informações sobre os porquês das decisões e ações tomadas, e não apenas um conjunto de informações desconexas. Por exemplo, depois que um diagnóstico de hipertensão foi confirmado, alguns medicamentos podem ser adicionados ou removidos do tratamento em consultas posteriores. Na representação do raciocínio do médico, esses medicamentos não podem estar desconexos, eles têm que estar relacionados com o diagnóstico de hipertensão. No modelo aqui proposto, considera-se que todos os diagnósticos e medicamentos ainda em curso devem ser mostrados cada vez que o gráfico de uma nova consulta é gerado. Por exemplo, se o paciente da primeira consulta volta para realizar uma nova consulta queixando-se de perda de apetite, dor muscular e dores de cabeça. O médico novamente mede a pressão do paciente, mede o nível de glicemia e a temperatura do paciente. Como os diagnósticos de hipertensão e diabetes já foram dados, o médico não os insere na base de dados de novo, essas informações já estão lá. Porém o médico descarta a possibilidade de dengue e diz que o paciente possui apenas um resfriado. Além disso, ele retira os medicamentos *metformina* e *hidroclorotiazida* e prescreve um novo chamado *propranolol*. O gráfico gerado automaticamente pelo sistema para a nova consulta é mostrada na figura 3. Quando esse gráfico é criado, as informações da consulta atual não apresentam nenhum tipo de relação, enquanto que as informações provenientes da consulta anterior já possuem algumas relações definidas. Todos os sintomas e exames são informações da consulta atual. Enquanto que os medicamentos e diagnósticos podem ser de consultas anteriores.

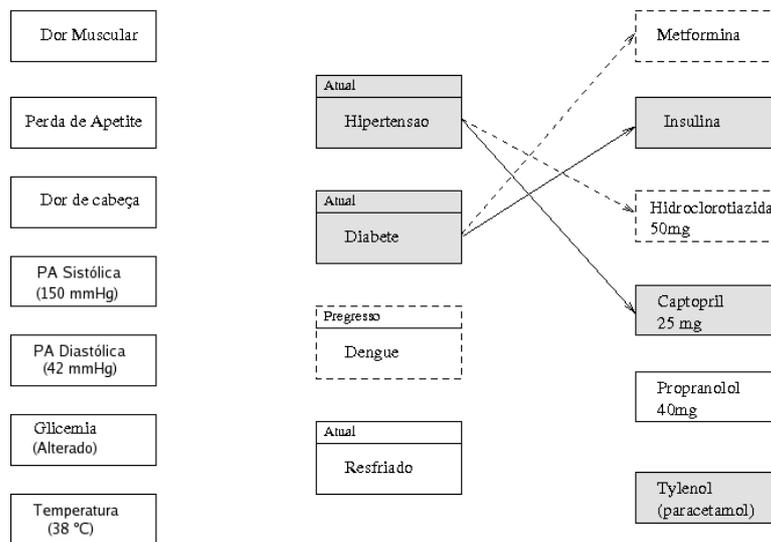
Quando um medicamento é retirado ou um diagnóstico se torna progresso, eles são representados por retângulos com as linhas pontilhadas e a partir da próxima consulta não irão mais aparecer no gráfico. Os diagnósticos e medicamentos das consultas anteriores são representados por retângulos mais escuros para serem diferenciados dos diagnósticos e medicamentos da consulta atual.



**Figura 02 - Gráfico gerado pelo sistema com as relações definidas pelo médico**

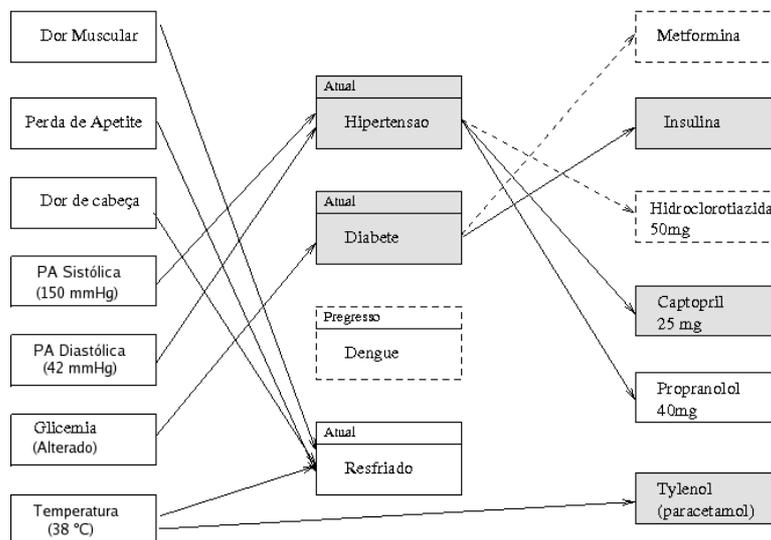
Na figura 3, o gráfico mostra que os diagnósticos de hipertensão e diabetes provêm da consulta anterior, enquanto que o diagnóstico de resfriado foi feito na consulta atual. Além disso, ele mostra que a suspeita de dengue não foi confirmada. O gráfico também mostra o que acontece com os medicamentos. Os que foram prescritos na consulta passada estão representados em retângulos mais escuros (*insulina*, *captopril* e *tylenol*), os que foram removidos estão em retângulos com linhas pontilhadas (*metformina* e *hidroclorotiazida*) o medicamento prescrito na consulta atual está em um retângulo mais claro. Como novas informações estão disponíveis no gráfico, o profissional de saúde deve definir as relações entre as informações. A figura 4 mostra um exemplo de como ficariam essas novas relações.

É possível observar na figura 4 que novamente a pressão arterial estava relacionada com o diagnóstico de hipertensão, assim como o exame de glicemia está associado ao diagnóstico de diabetes. Os outros sintomas e exames estão diretamente relacionados com o diagnóstico de resfriado, assim como o novo medicamento prescrito (*propranolol*) está ligado ao tratamento da hipertensão.

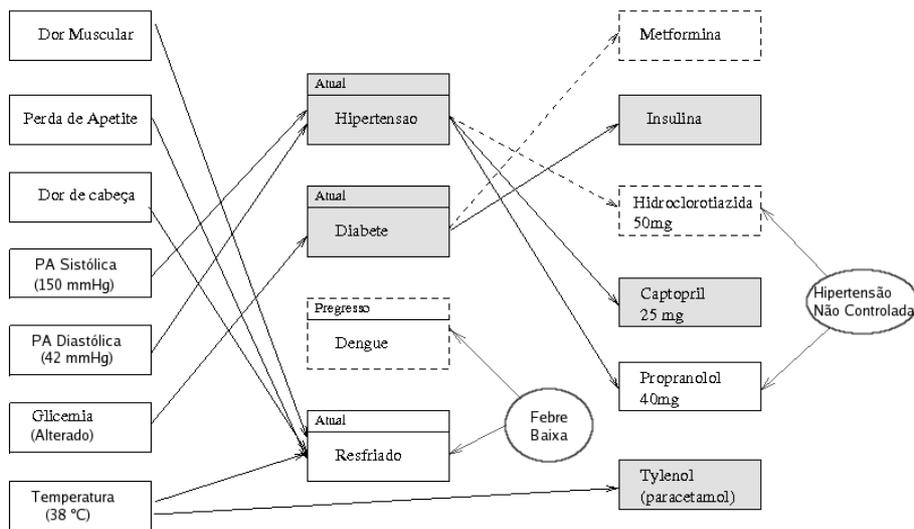


**Figura 03 - Exemplo do gráfico gerado em uma consulta de retorno**

Apesar das relações terem sido estabelecidas na figura 4, existem alguns pontos do raciocínio do profissional de saúde que ainda não ficaram claros. Por exemplo, “porque o diagnóstico de dengue foi retirado para a entrada do diagnóstico de resfriado?” ou “porque foi retirado o medicamento de *Hidroclorotiazida*?”. Existem perguntas que apenas o profissional de saúde que atendeu o paciente pode responder. Porém algumas delas podem ser respondidas com um simples comentário no gráfico da consulta. Continuando com o exemplo, o profissional de saúde inclui os comentários mostrados na figura 5.



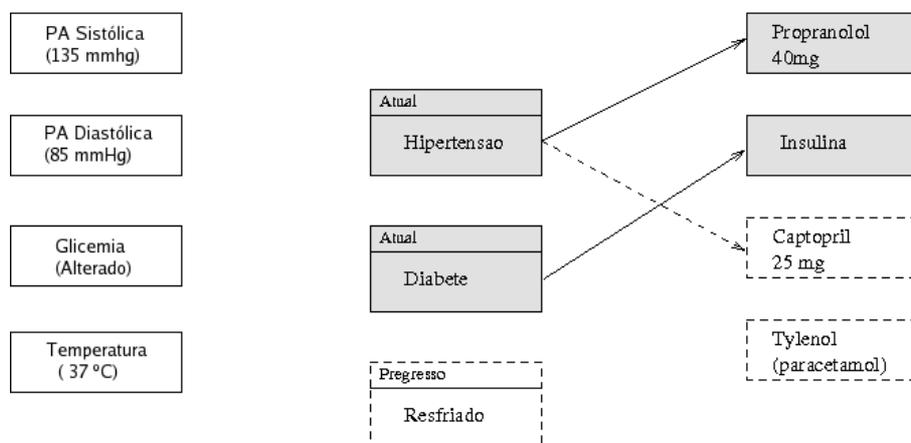
**Figura 04 - Exemplo de um gráfico de uma consulta de retorno com as relações definidas**



**Figura 05 - Exemplo do gráfico com relações e comentários**

De acordo com esses comentários, o profissional de saúde trocou o diagnóstico de dengue por resfriado, porque a febre do paciente não é alta. E ele faz a substituição no medicamento da hipertensão porque essa não foi controlada. Para elucidar um pouco mais o funcionamento do modelo, supõe-se que o paciente retorna para a terceira consulta. Mas agora o paciente não possui nenhuma queixa, diz estar se sentindo bem. Mesmo assim o profissional de saúde mede a sua pressão arterial, o nível de glicemia e a temperatura corpórea. Ele conclui que o resfriado passou, mas continua com os diagnósticos de hipertensão e diabete. Ele retira do tratamento do paciente os medicamentos *tylenol* e *captopril*.

A figura 6 mostra o gráfico da consulta atual. Novamente os retângulos mais escuros mostram as informações provenientes de consultas anteriores, os retângulos com linha tracejada indicam diagnósticos pregressos ou medicamentos retirados do tratamento e os retângulos mais claros as informações obtidas durante a atual consulta.



**Figura 06 - Exemplo do gráfico gerado na terceira consulta**

Como foi feito nas consultas anteriores, o profissional de saúde deve relacionar as informações para contextualizá-las no quadro clínico do paciente. A figura 7 mostra que a medição da pressão arterial foi relacionada com o diagnóstico de hipertensão, enquanto que o nível de glicemia foi relacionado com a diabete. Os comentários mostrados no gráfico da figura 7 justificam algumas das ações do profissional de saúde. Segundo os comentários, o diagnóstico de resfriado foi dado como progresso, já que o paciente não apresenta mais nenhum sintoma, inclusive a temperatura do paciente voltou ao normal. Além disso, existe a justificativa da retirada do *captopril* e a permanência do *propranolol* porque esse controlou a hipertensão, como é mostrado no exame de pressão arterial.

Através do exemplo mostrado, é possível perceber que a utilização de design rationale num ambiente clínico pode facilitar a compreensão das ações e decisões tomadas pelo profissional de saúde durante a consulta. E o modelo aqui proposto é uma forma de representar o raciocínio desse profissional.

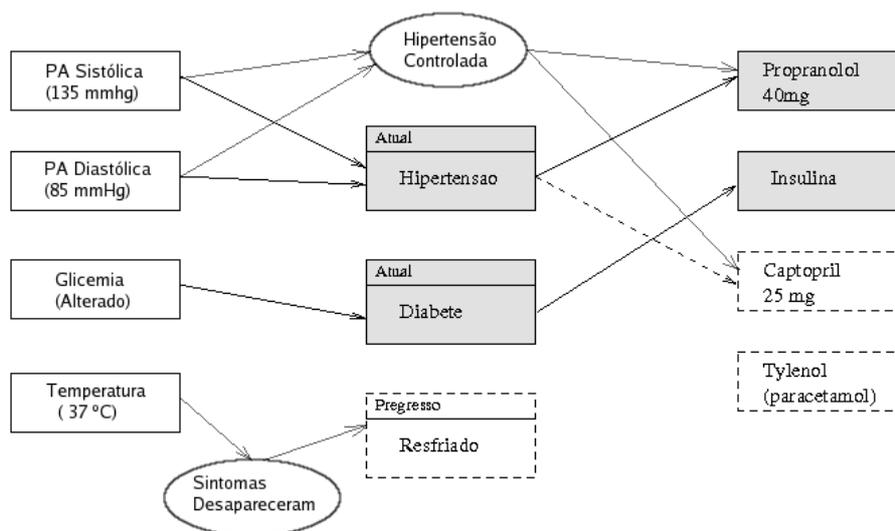


Figura 07 - Gráfico na terceira consulta com as relações definidas

## 5. Planos Para Avaliação do Modelo

Como proposta futura, pretende-se validar o modelo de design rationale e a introdução de um sistema colaborativo em um prontuário eletrônico. Será feito um experimento que avalie a qualidade, desempenho e aceitabilidade do modelo por um grupo de médicos.

Alguns dos critérios de avaliação serão: facilidade do uso do modelo para a representação do raciocínio; capacidade do modelo em representar um raciocínio, a facilidade do modelo para a compreensão do raciocínio e a satisfação dos usuários.

A facilidade de uso do modelo para a representação do raciocínio é de extrema

importância, porque o profissional de saúde não pode dispensar muito tempo na modelagem do raciocínio. O ideal é que o modelo seja intuitivo, para o profissional utilizá-lo sem a necessidade de grandes conhecimentos, ou treinamento. Já a capacidade do modelo para a representação do raciocínio está concentrada em manter o modelo simples e eficaz, capaz de representar o suficiente para que as ações do profissional de saúde sejam corretamente interpretadas. O critério de facilidade do modelo para a compreensão do raciocínio vai medir a facilidade dos profissionais de saúde em entenderem o raciocínio do profissional de saúde que realizou a consulta e construiu o “rationale” da consulta. Porém o critério mais importante é satisfação do profissional de saúde em utilizar o modelo, porque o que se deseja é que ele tenha vontade de demonstrar o seu raciocínio a fim de auxiliar no seu trabalho e no trabalho de seus colegas. O uso do modelo não pode se tornar uma obrigação imposta pelo sistema.

Para medir os critérios definidos, o sistema será disponibilizado em um ambulatório clínico da UNIFESP, onde profissionais de saúde terão acesso ao sistema. Esses profissionais receberão treinamento para utilização do sistema e serão observados durante a sua utilização. A validação do sistema será por meio de observação não participativa, entrevistas não estruturadas e entrevistas estruturadas, onde se espera poder observar a facilidade ou não uso do sistema, qual a interação com o sistema, e este sistema estimulou o profissional de saúde a descrever “os porquês” das suas decisões e se houve aumento de colaboração entre os médicos. Outro aspecto da análise destas entrevistas é captar informações que escapam da entrevista formal, como a empolgação do usuário em usar o sistema, comentários, críticas e sugestões sobre o modelo.

Ao final desta avaliação espera-se poder:

- Demonstrar que o modelo aqui proposto é capaz de representar de forma simples e eficaz o raciocínio do profissional de saúde para o registro de informações do paciente.
- Incentivar a utilização de “design rationale” na área de saúde para facilitar o acompanhamento do tratamento do paciente,
- Facilitar a comunicação entre os profissionais e
- Auxiliar o trabalho colaborativo na área da saúde.

## 7. Referencias

Alberdi, E., A. Povyakal, L. Strigini, M. Hartswood, R. Procter and R. Slack (2005): The use of Computer Aided Detection tools in screening mammography: A multidisciplinary investigation. British Journal of Radiology, special issue on Computer-aided diagnosis,

vol. 78: 31–40.

- Atkinson P. *Medical Talk and Medical Work: The Liturgy of the Clinic*, Sage, London (1995). London: Sage; 1995.
- Barsottini, C and Wainer J. Patterns of Collaboration and Non-Collaboration Among Physicians. Ed. Fuks H, Lukosch S., and Salgado A.C. 3706 ed. 2005, 248-54.
- Berg M. Accumulating and Coordinating: Occasions for Information Technologies in Medical Work. *Computer Supported Cooperative Work (CSCW)* 1999;8(4):373-401.
- Berg M. Patient care information systems and health care work: a sociotechnical approach. *International Journal of Medical Informatics* 1999 Aug;55(2):87-101.
- Berg, M., and Goorman, E., The contextual nature of medical information. *Int. J. Med. Inform.* 56:51–60, 1999.
- Clarke K, Hartswood M, Procter R, Rouncefield M. The Electronic Medical Record and Everyday Medical Work. *Health Informatics Journal* 2002;7((3/4)):168-70.
- Clarke, K., M. Hartswood, J. Hughes, R. Procter and M. Rouncefield (2006): ‘Normal, natural troubles: The practical organisation of bed management in a healthcare setting. In D. Francis and S. Hester (eds): *Orders of Ordinary Action: Respecifying Sociological Knowledge*. Ashgate Publishing.
- David pinelle and carl gutwin LOOSE COUPLING AND HEALTHCARE ORGANIZATIONS *Computer Supported Cooperative Work* (2006) 15:537–572
- Ellis CA, Gibbs SJ, Rein GL: Groupware some issues and experiences. *Communications of the ACM* 1991, 34: 38-58.
- Fussell, S.R., Kraut, R. E., Learch, F.J., Scherlis, W.L., McNally, M.M. and Cadiz, J.J. (1998) “Coordination, overload and team performance: effects of team communication strategies”, *Proceedings of CSCW '98*, Seattle, USA, p. 275-284.
- Gerosa, M.A., Fuks, H. and Lucena, C.J.P. (2001) “Use of Categorization and Structuring of Messages in Order to Organize the Discussion and Reduce Information Overload in Asynchronous Textual Communication Tools”, *Proceedings of the 7th International Workshop on Groupware - CRIWG*, Darmstadt, Germany, IEEE Computer Society, USA, p. 136-141.
- Hartswood M, Procter R, Rouncefield M, Slack R: Making a Case in Medical Work: Implications for the Electronic Medical Record. *Computer Supported Cooperative Work (CSCW)* 2003, 12: 241-266.
- Kraut, R.E. & Attewell, P. (1997), “Media use in global corporation: electronic mail and organizational knowledge”, in *Research milestone on the information highway*, Mahwah, NJ: Erlbaum, USA.
- Lorenzi, N. M., and Riley, R. T., Organizational impact of health information systems in healthcare. *Stud. Health Technol. Inform.* 65:396–406, 2002.
- MacLean A, Young RM, T.P.Moran TP. Design rationale: the argument behind the artifact.

SIGCHI Bull 2006;20(SI):247-52.

Örjan Dahlström, Ingrid Thyberg, Ursula Hass, Thomas Skogh, Toomas Timpka Designing a Decision Support System for Existing Clinical Organizational Structures: Considerations From a Rheumatology Clinic *Journal of Medical Systems*. 2006, Vol. 30, No. 5: 325

Pratt, W., Reddy, M. C., McDonald, D. W., Tarczy-Hornoch, P., and Gennari, J. H., Incorporating ideas from computer-supported cooperative work. *J. Biomed. Inform.* 37:128–137, 2004.

Strauss AL: *Social organization of medical work*. Chicago: University of Chicago Press; 1985.

Varjas, Kris, et al. "Using ethnographic methods for development of culture-specific interventions." *Journal of School Psychology* 43.3 (2005): 241-58.

Weerakkody G, Ray P,. CSCW-based system development methodology for health-care information systems. *Telemed J E Health*. 2003 Fall;9(3):273-82.

Xiao Y: Artifacts and collaborative work in healthcare: methodological, theoretical, and technological implications of the tangible. *Journal of Biomedical Informatics* 2005, 38: 26-33.

# Giga-Colab: Uma Arquitetura de Ambiente Virtual Colaborativo com suporte a Serviços Multimídia\*

Raoni Kulesza<sup>1</sup>, Luiz G. P. Alves<sup>1</sup>, Fernando L. Almeida<sup>1</sup>, Marcos F. Jardini<sup>1</sup>,  
Marco A. V. M. Vasconcelos<sup>2</sup>, Guido L. S. Filho<sup>2</sup>,  
Regina M. Silveira<sup>1</sup>, Graça Bressan<sup>1</sup>

<sup>1</sup>Laboratório de Arquitetura e Redes de Computadores (LARC) – Escola Politécnica da  
Universidade de São Paulo (EPUSP) – São Paulo – SP – Brasil

<sup>2</sup>Laboratório de Aplicações de Vídeo Digital (LAVID) – Departamento de Informática –  
Universidade Federal da Paraíba (UFPB) – João Pessoa – PB - Brasil

{raoni, luizgpa, falmeida, mjardini, regina, gbressan@larc.usp.br}  
{marcovas, guido@lavid.ufpb.br}

**Abstract.** *The development of collaborative and multimedia systems is a complex task and one of the key challenges is to promote the reuse and integration of those two software categories in the same environment. This paper presents an extensible collaborative virtual environment architecture based on client-server model, taking advantage of a service oriented approach and open multimedia standards (ISMA, H.323 and SIP). To validate this proposal, a prototype of a collaborative virtual environment was developed, supporting two multimedia applications: video on demand and videoconference. The project had as target the Giga Network (RNP/CPqD) and had as its main requirement the support of high definition video, resulting in some challenging issues to overcome.*

**Resumo.** *Desenvolver sistemas colaborativos e multimídia é uma tarefa bastante complexa e um dos principais desafios é promover a reutilização e integração dessas duas categorias de software num mesmo ambiente. Neste contexto, apresentamos uma arquitetura de um ambiente virtual colaborativo extensível baseada no modelo cliente-servidor, mas que utiliza uma abordagem orientada a serviços e padrões abertos da área multimídia (ISMA, H.323 e SIP). Para validar esta proposta, foi desenvolvido um protótipo de um ambiente virtual colaborativo com suporte a duas aplicações multimídia: vídeo sob demanda e videoconferência. O projeto é focado na Rede Giga (RNP/CPqD) e tem como principal requisito o suporte a vídeos de alta definição, resultando em alguns problemas desafiadores a serem superados.*

## 1. Introdução

Sistemas de suporte ao trabalho cooperativo (do inglês CSCW, *Computer Supported Cooperative Work*), ou sistemas colaborativos, incluem um ambiente compartilhado com funcionalidades relacionadas à cooperação, coordenação e comunicação para suportar grupos de usuários engajados em uma atividade comum (ou objetivo comum).

---

\* Trabalho Financiado pela RNP com recursos FINEP/FUNTTTEL (Sub-Projeto Giga 2472)

Devido ao avanço nas áreas de computação e redes comunicação, cada vez mais esses sistemas utilizam aplicações multimídia com o objetivo de melhorar as funcionalidades e experiência dos usuários no processo de colaboração [Steinmetz e Nahrstedt 2004]. Com isso, o desenvolvimento de sistemas colaborativos, que já era considerado complexo [Ellis et al. 1991], torna-se um desafio ainda maior, uma vez que as aplicações multimídia são caracterizadas pelo processamento de grande volume de dados e tempo de reprodução contínuo. Adicionalmente, devido à natureza dinâmica das funcionalidades colaborativas, também é exigido que os sistemas colaborativos sejam flexíveis o suficiente para se adequarem a novos requisitos do ambiente e usuários [Kahler et al. 2000].

Para tratar o aspecto da complexidade na construção de sistemas colaborativos esta proposta se baseia em um Ambiente Virtual Colaborativo extensível (do inglês *CVE, Collaborative Virtual Environment*) desenvolvido por [Harada 2001], que fornece uma infra-estrutura para o desenvolvimento de novas aplicações colaborativas: o *jVE (Java Virtual Environment)*. Entretanto, de forma a facilitar a integração de novas funcionalidades e interoperabilidade com aplicações já existentes, a arquitetura do ambiente foi estendida para suportar tecnologias de serviços Web e protocolos abertos na área de sistemas multimídia. Particularmente, foram utilizados SOAP (*Simple Object Access Protocol*) e WSDL (*Web Service Definition Language*) [Alonso et al. 2004] para acesso e configuração de serviços, os padrões ISMA (*Internet Streaming Media Alliance*) [Fuchs e Farber 2005] para distribuição de mídias, e H.323 [H.323 1999] e SIP (*Session Initiation Protocol*) [Rosenberg et al. 2002] para controle de sessões de conferência.

Como principal resultado deste trabalho, uma solução de um ambiente virtual colaborativo está sendo desenvolvida no contexto do projeto GIGA-RNP, de forma a facilitar a interação e o trabalho cooperativo entre pessoas situadas remotamente através de conferência multimídia e transmissão e recepção de elevado volume dados. Uma grande vantagem desta proposta é a reutilização de aplicações multimídia já desenvolvidas anteriormente em outros projetos de nossos grupos e que têm capacidades elevadas para a transmissão e recepção de vídeo.

Além desta seção de introdução, este artigo está estruturado em mais 4 seções. A próxima seção apresenta os trabalhos relacionados. Na seção 3 é apresentada a proposta, através de uma análise, visão geral da arquitetura e detalhamento dos principais módulos. A seção 4 descreve a implementação de um protótipo e resultados preliminares da implantação na rede do projeto GIGA-RNP. Na seção 5 são discutidas as conclusões e trabalhos futuros e em andamento.

## **2. Trabalhos Correlatos**

Atualmente, existem vários sistemas colaborativos na comunidade científica que fornecem suporte a aplicações colaborativas e multimídia conjuntamente. Destacam-se os projetos *Agora* [Agora], do “*Centre for e-Science*” da Universidade de Lancaster, *AccessGrid* [AccessGrid], do “*Argonne National Laboratory*” de Chicago, o *ConferenceXP* [Beavers et al. 2004], da *Microsoft Research* e o *Global Multimedia Collaboration System* (Global MMCS) [Wu et al. 2004] do “*Communit Grids Lab*” da Universidade de Indiana.

O *AccessGrid*, baseia-se numa infra-estrutura de servidores espalhados, chamados de nós, interconectados formando uma grade. O acesso à grade e os requisitos para se estabelecer uma sessão colaborativa seguem um protocolo próprio (o *AccessGrid* ou *AG*). O *AG* define um mecanismo de comunicação entre todas as entidades do sistema. É através dele que se inicia uma nova sessão para o ambiente colaborativo, chamada de “local do evento ou fonte de vídeo” (do inglês, *Venue*) e definem-se as características e recursos utilizados pela mesma, como portas no servidor e as capacidades de áudio e vídeo.

O *ConferenceXP* é semelhante ao *AccessGrid*. A principal diferença entre ambos está no emprego dos protocolos de comunicação, enquanto o segundo baseia-se na utilização restrita de aplicações multimídia existentes (*VIC* e *RAT*)<sup>1</sup> sobre vários protocolos (*RTSP*, *RTP*, *XMPP*, *AG*) e uma rede ponto-a-ponto de acesso restrito (os servidores de *Venue*), o primeiro utiliza somente *RTP* para todas as comunicações e o conceito de rede ponto-a-ponto com acesso através da Internet, ou seja, não restrito, para tratar as interligações entre os participantes. Adicionalmente, o *ConferenceXP* é desenvolvido totalmente sobre a plataforma .NET, limitando a portabilidade do sistema. Ressalta-se a utilização de um mecanismo específico e simplificado de comunicação, que define um pacote genérico de transporte chamado *RTOjects*. Estes são objetos (do inglês, *Objects*) usados para fluxo de dados e troca de mensagens e que, apesar de serem encapsulados em pacotes *RTP*, são diferenciados dos fluxos de áudio e vídeo.

O *Agora* é uma ferramenta baseada na plataforma Java, que procura criar um ambiente colaborativo completo, abrangendo as mais diversas formas de comunicação e colaboração. Ele é desenvolvido sobre o ambiente *SAKAI*, seguindo especificações *Java Enterprise Edition*<sup>2</sup> (*servlets* e *portlets*), e gerencia todo o processo de uma sessão de colaboração, oferecendo aplicações colaborativas. O *Agora* engloba as funções do *SAKAI* acrescentando uma interface cliente personalizável com suporte conferência multimídia utilizando o *Java Media Framework*<sup>3</sup> (*JMF*), área de trabalho compartilhada, *whiteboard* e vídeo sob demanda.

Seguindo o conceito de descentralização que as redes ponto-a-ponto oferecem como no *ConferenceXP*, o *Global MMCS* oferece um sistema distribuído, baseado em serviços Web, para gerenciamento de sessões de colaboração e com suporte a multimídia. Como infra-estrutura de distribuição é utilizada uma rede sobreposta, que é capaz de registrar e buscar informações dos participantes de forma interoperável. O principal objetivo é tratar de forma transparente a heterogeneidade de sistemas existentes, possibilitando o uso dos protocolos *SIP*, *H.323* e *AG* para sessões de conferência. Assim como o *Agora*, o *Global MMCS* oferece um cliente *JMF* com suporte de áudio e vídeo.

A principal diferença da arquitetura Giga-Colab, que será apresentada neste trabalho, em relação aos demais trabalhos com escopo similar, é a utilização de uma arquitetura híbrida, cliente-servidor para as aplicações de colaboração e baseada em serviços (que pode se tornar ponto-a-ponto) para as aplicações multimídia. Outro

---

<sup>1</sup> Soluções de software-livre descontinuadas e limitadas no suporte a áudio e vídeo.

<sup>2</sup> Solução da *Sun Microsystems*<sup>TM</sup> para desenvolvimento de aplicações corporativas avançadas.

<sup>3</sup> Solução da *Sun Microsystems*<sup>TM</sup> para desenvolvimento de aplicações multimídia.

aspecto é a capacidade de tratamento de vídeo. No Giga-Colab as aplicações multimídia são baseadas em soluções desenvolvidas em código nativo integrado com Java e suporte a padrões recentes de codificação, o que oferece um melhor desempenho e capacidade na exibição de vídeos para conferência e/ou vídeo sob demanda. A Tabela 1 apresenta um resumo comparativo das características de cada trabalho.

**Tabela 1. Comparação de características entre ambientes colaborativos**

	<b>AccessGrid</b>	<b>ConferenceXP</b>	<b>Agora</b>	<b>Global MMCS</b>	<b>Giga-Colab</b>
<b>Tipo de Arquitetura para Gerenciamento de Sessão Colaborativa</b>	<i>Ponto-a-ponto (para servidores de Venue)</i>	<i>Ponto-a-ponto</i>	<i>Cliente-Servidor</i>	<i>Ponto-a-ponto</i>	<i>Cliente-Servidor</i>
<b>Bibliotecas para áudio e vídeo</b>	<i>Nativas (VIC e RAT)</i>	<i>DirectShow e Windows Media Player</i>	<i>Java Media Framework</i>	<i>Java Media Framework</i>	<i>Java2D + Nativas (pwlif e ffmpeglib)</i>
<b>Suporte a ferramentas de colaboração</b>	<i>Através de desenvolvimento de terceiros</i>	<i>Integração com qualquer produto .NET</i>	<i>WhiteBoard, chat de texto, área de trabalho compartilhada. Capacidade de integrar serviços web</i>	<i>Em desenvolvimento Capacidade de integrar serviços web</i>	<i>WhiteBoard e chat de texto. Integração com qualquer aplicação Java (através do jVE) ou serviços web</i>
<b>Capacidade de Gravação de Vídeo</b>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>
<b>Resoluções de vídeo suportadas</b>	<i>176x144, 320x240, 352x288</i>	<i>Informação não fornecida</i>	<i>176x144, 320x240, 352x288</i>	<i>176x144, 320x240, 352x288</i>	<i>176x144, 352x288, 720x480, 1280x720</i>
<b>Multiplataforma</b>	<i>Sim</i>	<i>Não, apenas Windows</i>	<i>Sim, mas com limitações no Linux para vídeo.</i>	<i>Sim, mas com limitações no Linux para vídeo.</i>	<i>Sim</i>
<b>Número de participantes em áudio e vídeo<sup>4</sup></b>	<i>Ilimitado</i>	<i>4</i>	<i>Ilimitado<sup>3</sup></i>	<i>Ilimitado<sup>3</sup></i>	<i>16</i>
<b>Suporte a protocolos abertos na área de aplicações multimídia</b>	<i>RTSP e RTP</i>	<i>RTP</i>	<i>RTP</i>	<i>SIP, H.323, AG</i>	<i>HTTP, RTSP, SDP, RTP, SIP e H.323</i>

### 3. Arquitetura Giga-Colab

Nesta seção serão apresentadas inicialmente (subseção 3.1) as características (requisitos) do domínio de aplicações de CSCW que o Giga-Colab deverá atender incluindo características comuns aos projetos analisados. As demais subseções (3.2 e 3.3) descrevem uma proposta de arquitetura para sistemas colaborativos, apresentando seus elementos e as principais interfaces de comunicação entre eles.

<sup>4</sup> Quando ilimitado, o número de participantes depende da largura de banda disponível e capacidade de processamento das máquinas dos clientes.

### 3.1. Características

As características comuns de sistemas colaborativos foram identificadas através da descrição de dimensões de análise propostas por [Licea 2006] e do modelo 3C de Colaboração apresentado por [Gerosa 2006].

De acordo com [Licea 2006] há sete dimensões de análise obtidas do estudo de sistemas colaborativos, de modo que cada dimensão representa um requisito ou decisão de projeto para a aplicação. As dimensões são: interação, participação, visualização, notificação, coordenação, distribuição e suporte. Já [Gerosa 2006] faz uma análise do modelo 3C de Colaboração proposto originalmente por [Ellis et al. 1991] para suporte computacional à colaboração e que se baseia em três principais características: comunicação, coordenação e cooperação. A percepção (do inglês, *awareness*), que permeia os 3C's, também é um requisito essencial presente em sistemas colaborativos. A partir do estudo dos trabalhos citados, foram destacados os requisitos para compor a arquitetura Giga-Colab:

- Existência de um ambiente virtual que representa um espaço compartilhado no qual há operação conjunta dos usuários para atender a cooperação;
- Definição de uma arquitetura de comunicação, que neste caso é caracterizada como cliente-servidor, uma vez que o jVE é baseado na relação entre clientes e um ambiente virtual centralizado;
- As informações do ambiente virtual devem ser compartilhadas através de interação síncrona ou assíncrona, e explícita ou implicitamente. A troca de informações é dita explícita quando um cliente inicia explicitamente o envio da informação e implícita quando o sistema dissemina uma informação sem que um cliente precise solicitar;
- Definição de um modelo para coordenação do ambiente virtual: uma Sala (metáfora de uma sala real voltada para um ambiente virtual) pode conter uma ou mais atividades associadas. Cada atividade é composta por usuários com suas respectivas permissões, um repositório compartilhado de conteúdo (arquivos de vários tipos de mídias, representando documentos, imagens, áudio e vídeo) e aplicações colaborativas (ver seção 4). A coordenação também abrange aspectos de controle de concorrência no acesso e, isto é realizado no nível de objetos do ambiente virtual;
- Existência de um mecanismo de persistência da informação, armazenando o estado dos objetos de uma Sala. Adicionalmente, é necessário também um mecanismo de *cache* desses objetos, de forma a melhorar a recuperação dessa informação;
- Suporte a um mecanismo de notificação e percepção das atividades do grupo, onde o sistema comunica aos usuários mudanças no contexto de uma Sala (entrada/saída de usuário, inclusão de novo arquivo no repositório, estado atual da tarefa realizada etc.);
- Ser extensível para suportar a integração com ferramentas e serviços externos;
- Independência de interface de usuário.

De posse destes requisitos elencados acima será detalhado na próxima subseção a visão geral da arquitetura proposta.

### 3.2. Visão Geral

A estrutura geral da arquitetura proposta neste trabalho (Figura 1) é baseada em um modelo híbrido, onde existe uma entidade central, denominada *Servidor Giga-Colab* (responsável pelo gerenciamento da colaboração e dos serviços) e várias outras entidades, denominadas *Serviços*, que podem estar distribuídas pela rede e complementam as funcionalidades de colaboração, provendo serviços tanto para o servidor central, como diretamente para os clientes (*Cliente Giga-Colab*).

A arquitetura concebida permitiu a reutilização de módulos do jVE desenvolvidos por [Harada 2001] que se beneficiam do modelo cliente-servidor e ao mesmo tempo, através do uso de tecnologias de serviços Web, provê flexibilidade e escalabilidade para adicionar novas funcionalidades, por exemplo, para as aplicações multimídia servidoras. Nas subseções seguintes serão detalhados cada subsistema da arquitetura e seus respectivos módulos.

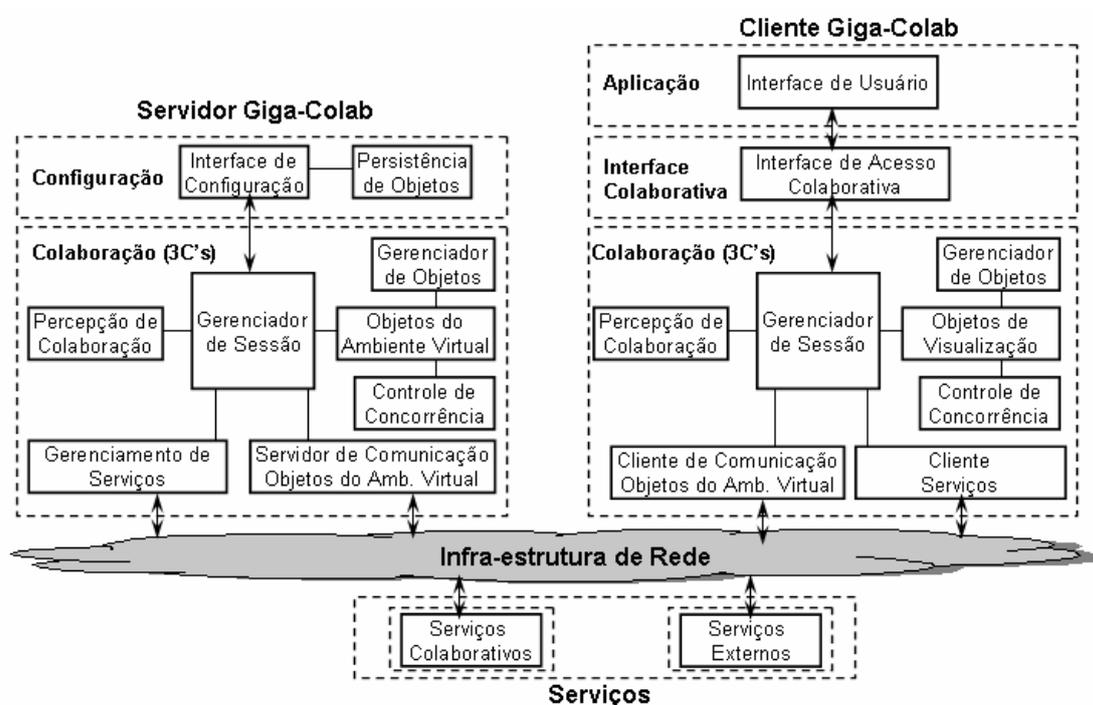


Figura 1. Arquitetura Geral da Solução

#### 3.2.1. Servidor Giga-Colab (GC)

O Servidor GC representa a entidade capaz de fornecer um conjunto de funcionalidades de colaboração (subsistemas *Colaboração e Configuração*) que permite o acesso remoto do Cliente GC ao ambiente virtual. Abaixo uma descrição de cada módulo:

- **Interface de Configuração** – define uma interface de acesso para administração de todas as entidades do ambiente virtual colaborativo, definindo um modelo de dados e mecanismos de acesso para o mesmo. Este módulo é responsável pela pré-articulação das tarefas, ou seja, pelas ações necessárias para preparar a colaboração, normalmente concluídas antes do trabalho colaborativo se iniciar.

- Persistência de Objetos – representa a base de dados para armazenamento do estado dos objetos do ambiente virtual colaborativo e também das entidades relacionadas a estes objetos. Consiste na pós-articulação das tarefas ocorrida ao seu término.
- Gerenciador de Sessão – provê funcionalidades que permitem que o servidor GC gerencie dinamicamente os participantes de uma Sala, bem como a troca de dados e operações entre as entidades associadas. Para isso cria uma sessão colaborativa para cada Sala e controla todas as suas associações (Atividades, Usuários, Ferramentas, Repositórios e Serviços). Este módulo é responsável também por realizar o controle de acesso de cada participante através de um mecanismo de autenticação e autorização.
- Controle de Concorrência – podem ocorrer problemas relacionados à consistência, pois os estados dos objetos são replicados e passíveis de alterações de várias entidades de forma concorrente. A estrutura fornecida para o controle de concorrência é feita no nível de objetos através do mecanismo de bloqueios. Este controle é implementado na raiz da hierarquia de classes do ambiente virtual, sendo assim herdado por todas as outras classes. O controle por bloqueio funciona como um semáforo, pois para poder realizar uma alteração deve-se possuir o bloqueio da propriedade sendo alterada, ou seja, quando um usuário desejar, por exemplo, utilizar uma aplicação, ele deve inicialmente fazer o pedido de bloqueio de uma determinada propriedade para em seguida realizar o trabalho colaborativo, caso ele tenha permissão de efetuar tal operação. O servidor GC também prevê um outro controle de concorrência que utiliza o conceito de transformadas operacionais [Harada 2001]. Com este mecanismo é possível, por exemplo, que duas pessoas editem simultaneamente o mesmo texto, tornando-se útil quando o tempo de bloqueio é intolerável.
- Objetos do Ambiente Virtual – são entidades que habitam o ambiente virtual e possuem: (i) comportamento – associado à capacidade de gerar e responder eventos que resultam na execução de alguma tarefa e; (ii) propriedades – são responsáveis por armazenar o estado do objeto (identificação, localização, conteúdo).
- Gerenciador de Objetos – define as funcionalidades para os participantes salvarem os objetos no repositório e recuperá-los em uma próxima sessão. Este elemento é responsável por manter o ambiente virtual ativo, pois o Gerenciador de Objetos mantém um buffer contendo todos os objetos do ambiente virtual em uso no servidor (mantidos na memória). Quando um objeto do ambiente virtual é requisitado, este módulo é encarregado de localizar a classe que o implementa, instanciá-lo garantindo que não existam réplicas do objeto e recuperar os dados persistentes do objeto (dados que foram armazenados quando o objeto foi descarregado pela última vez).
- Percepção de Colaboração – permite que os participantes estejam cientes da presença de outros participantes na sessão e, além disso, ter a noção da atividade do grupo e do estado atual da tarefa. Os participantes podem ser representados por meio de uma lista, ícones ou então barra de status.
- Servidor de Comunicação Objetos do Ambiente Virtual – inclui funcionalidades responsáveis pelo tratamento da conexão e de eventos recebidos dos participantes. Consiste na implementação do protocolo de comunicação da colaboração executada entre os participantes conectados no ambiente virtual. Pode-se dizer que corresponde à execução do servidor colaborativo propriamente dito. Tem a função de enviar a

replicação das informações compartilhadas do ambiente virtual. A comunicação dá-se, basicamente, de duas formas: (i) entre objetos do ambiente virtual; (ii) entre objetos do ambiente virtual e os objetos de visualização correspondentes que estão localizados nos clientes. Todas as mensagens enviadas de e para os clientes são assíncronas. As mensagens enviadas de um objeto do ambiente virtual para um objeto de visualização normalmente transportam atualizações da representação para todos aqueles que estão no mesmo contexto de tarefa. Já as mensagens enviadas de um objeto de visualização para um objeto do ambiente virtual geralmente correspondem a uma ação do usuário.

- Gerenciamento de Serviços – este módulo possui uma interface de comunicação para gerenciar os serviços registrados no ambiente virtual colaborativo utilizando WSDL e SOAP. Pode realizar operações para sincronizar o serviço remoto de acordo com mudanças de configuração ou estado do ambiente virtual, assim como requisição de dados para objetos do ambiente virtual.

### 3.2.2. Cliente Giga-Colab (GC)

Representa os usuários finais do ambiente virtual colaborativo. O cliente Giga-Colab possui três subsistemas, sendo que o subsistema Colaboração é semelhante ao do servidor, mas executando funcionalidades relacionadas a um cliente e com representação dos objetos do ambiente virtual (módulo Objetos de Visualização), uma vez que uma instância de um objeto existe apenas no servidor e é replicado para os clientes.

O subsistema Interface Colaborativa tem o propósito de fornecer uma API (*Application Program Interface*) para possibilitar a integração de aplicações Java no ambiente colaborativo empregando o paradigma WYSIWIS (*What-You-See-Is-What-I-See*) [Borghoff e Schlichter 2000]. Adicionalmente, possui mecanismos específicos para notificação de eventos e mensagens do ambiente de colaboração que são úteis para o desenvolvimento de interfaces gráficas especializadas, destinadas exclusivamente à colaboração como uma janela de inspeção, por exemplo. E, por fim, o subsistema Aplicação compreende a aplicação colaborativa a ser acoplada no cliente Giga-Colab.

### 3.2.3. Serviços

No contexto da arquitetura proposta, um serviço representa uma aplicação que fornece uma interface de acesso remoto utilizando tecnologias de serviços *Web*. Esta interface pode oferecer tanto acesso às funcionalidades do serviço (por exemplo, acesso a dados), como também acesso a funcionalidades de gerenciamento (por exemplo, envio de comando). Existem dois tipos de serviços utilizados na arquitetura: Colaborativo e Externo. O primeiro representa uma aplicação que disponibiliza dados para o *Servidor GC* complementar as funcionalidades do ambiente virtual colaborativo e, como exemplo, pode-se citar uma aplicação científica que fornece dados de forma transparente para uma ferramenta exibir gráficos estatísticos no ambiente colaborativo. É interessante notar que o *Servidor GC* não conhece os aspectos de implementação da aplicação científica citada no exemplo anterior, ele apenas acessa a interface de serviço provida, o mesmo acontece com aplicações de outra natureza. O segundo não tem ligação direta com as funcionalidades do ambiente colaborativo sendo que o *Cliente GC*, através de parâmetros de configuração das entidades, acessa-o de forma independente. Uma ilustração desse tipo de serviço é a requisição de um arquivo de

vídeo (representado por uma entidade do repositório) em um Cliente de Serviço do Cliente GC para um Serviço de vídeo sob demanda.

### 3.3. Funcionamento

Com o objetivo de explicar a integração dos serviços com o ambiente virtual, as figuras 2a, 2b e 2c representam alguns cenários de funcionamento da arquitetura proposta.

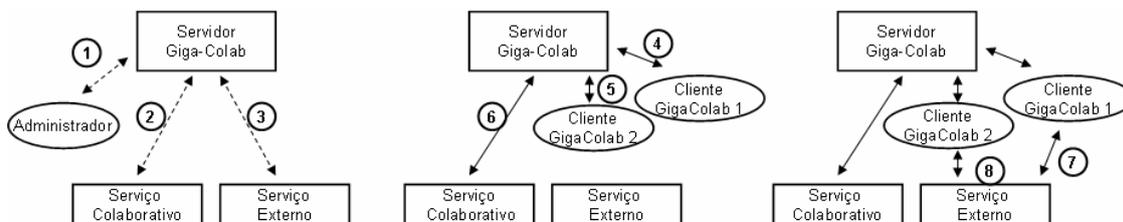


Figura 2a. Cenário 1

Figura 2b. Cenário 2

Figura 2c. Cenário 3

No Cenário 1 (Figura 2a) um usuário Administrador cadastra um Serviço Colaborativo e um Serviço Externo (seta 1). A partir dessa configuração o Servidor GC pode realizar operações de gerenciamento (por exemplo, sincronizar configurações ou passar parâmetros de acesso para os Clientes GC) ou acessar dados dos serviços (setas 2 e 3). Neste cenário todas as mensagens utilizam SOAP. No Cenário 2 (Figura 2b) é descrito o acesso de dois Clientes GC aos dados de um Serviço Colaborativo (setas 4 e 5), através do Servidor GC (seta 6) e as configurações realizadas pelo administrador no cenário anterior. Neste caso, para melhorar o desempenho da transmissão de dados, é possível utilizar qualquer protocolo de transporte, é necessário apenas o suporte do mesmo nas entidades que irão receber os dados. No último cenário (Figura 2c) é ilustrado o acesso dos Clientes GC a um Serviço Externo (setas 7 e 8), é possível notar que não é necessária uma comunicação com o Servidor GC, uma vez que não existe colaboração e os parâmetros de acesso aos Serviços Externos já foram repassados para os Clientes GC no momento da conexão com o Servidor GC (Cenário 2).

## 4. Protótipo do Ambiente Virtual Colaborativo

Com o objetivo de validar a arquitetura descrita na seção anterior, está sendo implementado um protótipo do Ambiente Virtual Colaborativo no contexto do Projeto GIGA-RNP. Até o momento, foram desenvolvidos os seguintes subsistemas: (i) Portal Web de Administração (configuração e administração), (ii) Servidor GC com principais módulos, (iii) Serviço Multimídia de Vídeo sob demanda, (iv) Serviço Multimídia de Videoconferência e (v) Cliente GC com seus principais módulos. Nas próximas subseções será apresentada uma breve descrição de alguns desses subsistemas.

Em relação às opções de tecnologias foram escolhidas soluções de código aberto escritas em C++ para os módulos de transmissão e recepção multimídia e a plataforma de desenvolvimento Java para as demais funcionalidades. Entretanto, para o caso do subsistema do Cliente GC foi realizada uma integração entre as funcionalidades de C++ e Java através da tecnologia *Java Native Interface* (JNI).

### 4.1 Portal de Administração

O Portal de Administração permite ao administrador do sistema configurar informações pertinentes ao contexto do sistema de colaboração (pré-articulação das tarefas). Através

de um *browser* o administrador realiza o gerenciamento (cadastro, edição, remoção etc.) das principais entidades do sistema (salas, atividades, repositórios, ferramentas e serviços), seus relacionamentos e o controle de acesso aos recursos.

Como principais funcionalidades, podem-se descrever o gerenciamento de salas e atividades, que corresponde à configuração do sistema através do cadastro dessas duas entidades e associação dos usuários, repositórios de conteúdo, ferramentas e serviços relacionados a uma determinada atividade, além dos repositórios disponíveis no sistema colaborativo. Uma atividade corresponde a uma tarefa específica que estará associada a uma única sala. A interface Administração de Atividades permite ao administrador visualizar e administrar informações referentes às atividades associadas em uma determinada sala, por exemplo, nome da atividade, sala associada, data e hora de início e fim, descrição, participantes desta atividade, ferramentas utilizadas e repositórios disponíveis. As interfaces do Portal de Administração que permitem a manipulação de informações das Salas e Atividades podem ser visualizadas na Figura 3.

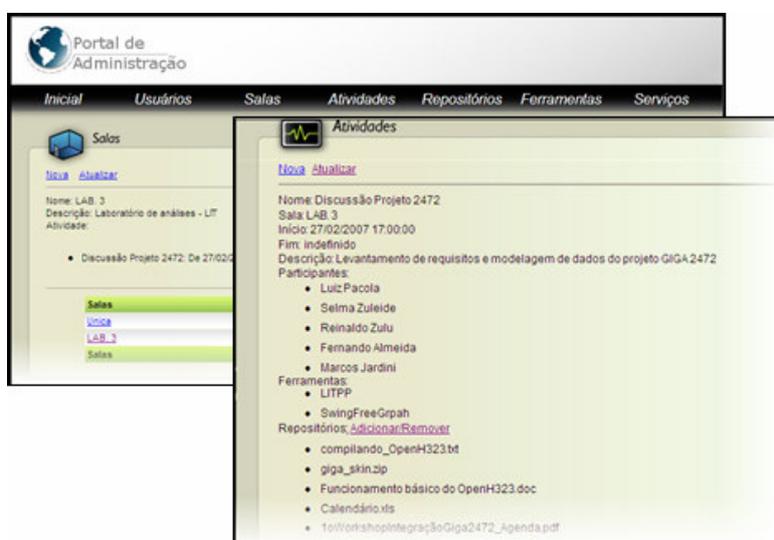


Figura 3. Interfaces Portal de Administração – Salas e Atividades

## 4.2 Serviços Multimídia

Os serviços multimídia desenvolvidos no protótipo utilizam duas aplicações desenvolvidas em trabalhos anteriores do grupo<sup>5</sup> e permitem que usuários do ambiente virtual colaborativo visualizem sob demanda arquivos multimídia e realizem comunicação síncrona através de canais de vídeo, áudio e texto. Uma visão geral desses subsistemas é ilustrada na Figura 4 e a seguir são descritos resumidamente cada módulo:

- Cliente JSC – representa o módulo responsável pela recepção e exibição de arquivos multimídia disponibilizados pelo Servidor JSC, ou seja, um cliente de vídeo sob demanda. Executa em plataforma *Windows* e *Linux*, utiliza a biblioteca *ffmpeg* [FFmpeg] para tratamento multimídia na recepção (transporte, demultiplexação,

<sup>5</sup> *Java Stream Control (JSC) para vídeo sob demanda e Dynavideo Conference System (DCS)* [Vasconcelos e Filho 2004].

decodificação e exibição), a *API JSA*<sup>6</sup> [JSA 2004] para estabelecimento e controle das sessões de recepção (através de HTTP e RTSP) e bibliotecas padrão do J2SE para implementar a interface gráfica.

- Servidor JSC – módulo que implementa o armazenamento, recuperação e transmissão de arquivos multimídia representando um servidor de vídeo sob demanda. Executa apenas em ambiente *Linux* e utiliza a biblioteca *liveMedia* [LiveMedia] para tratamento multimídia na transmissão (leitura, multiplexação e transporte), a API JSA para gerenciamento das sessões de transmissão com os clientes e bibliotecas da Apache Axis para configuração remota com tecnologias de serviços Web [Axis].
- Cliente DCS – módulo que representa um ponto de comunicação em um sistema de conferência, permitindo a transmissão e recepção de vídeo, áudio e texto com os demais usuários. Sua implementação possui uma interface gráfica e de controle em Java e é baseada no projeto *Opal* [VoxGratia]<sup>7</sup>, que implementa o protocolo H.323 e SIP em C++ e com a utilização da biblioteca *PwLib* [VoxGratia] é portátil para plataformas *Linux* e *Windows*.
- Servidor DCS – módulo que implementa um servidor para vídeo conferência centralizada baseada no protocolo H.323 e com suporte a serviços de um gerenciador de chamadas (*Gatekeeper* ou GK) e uma unidade de controle multiponto (*Multipoint Control Unit* ou MCU). O servidor executa em ambiente *Linux* utilizando aplicações (*GnuGk* e *OpenMCU*, respectivamente para o GK e a MCU) e bibliotecas (*OpenH323* e *PwLib*) desenvolvidas em C++ no projeto *VoxGratia*. Adicionalmente, foram desenvolvidas aplicações para controle remoto dos dois serviços com Apache Axis.

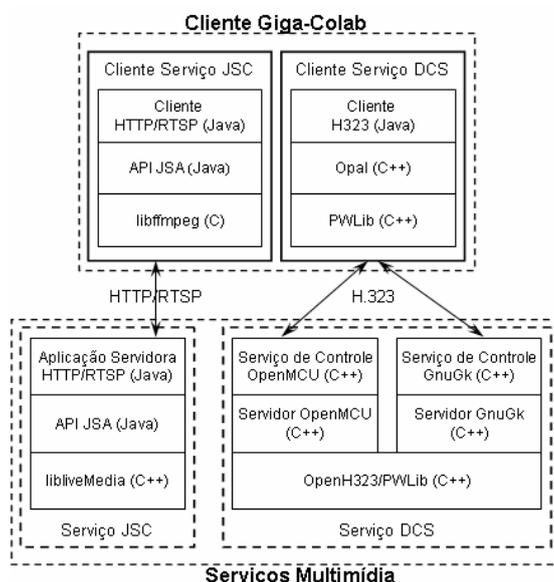


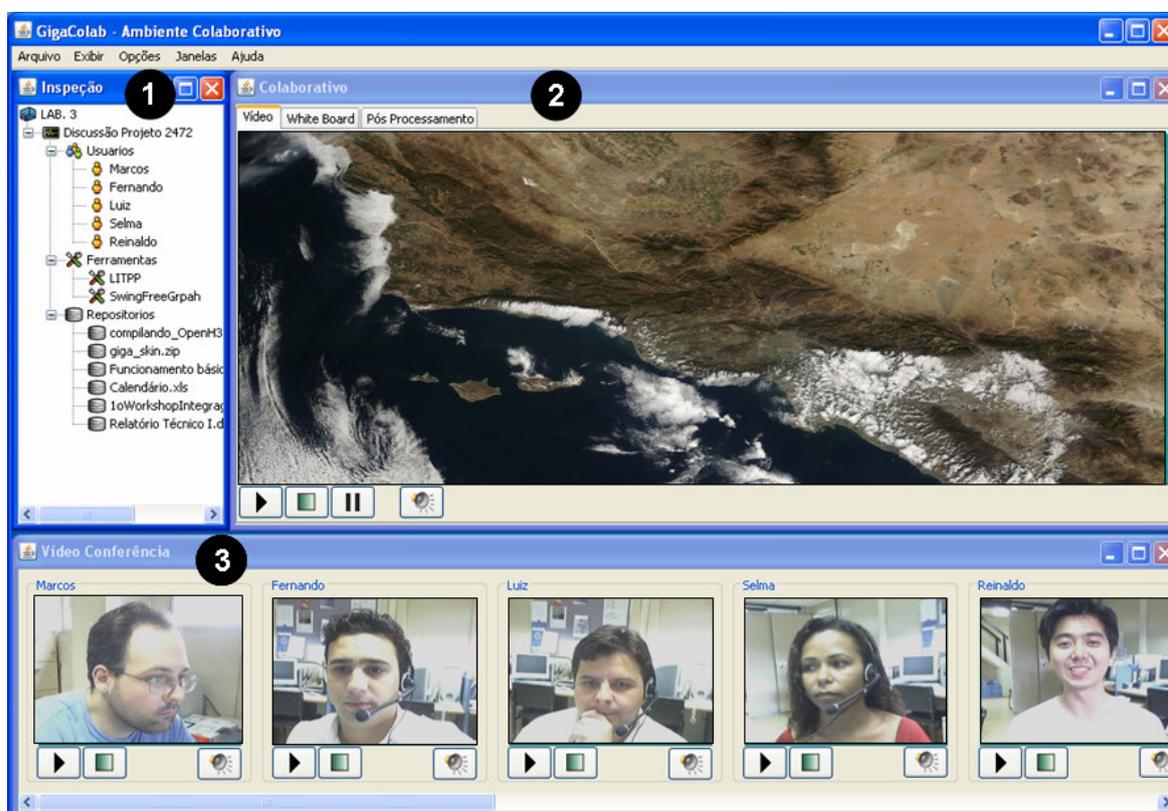
Figura 4 - Implementação de Serviços Multimídia

### 4.3. Interface Gráfica do Cliente Giga-Colab

<sup>6</sup> A JSA é tem como objetivo fornecer para aplicações multimídia em rede (particularmente, no lado do servidor) uma interface de software unificada e padronizada para gerenciamento de fluxos de mídias.

<sup>7</sup> VoxGratia é um projeto de código-fonte aberto com implementações dos padrões SIP e H.323.

O Cliente GC do protótipo apresenta uma interface gráfica em Java para os usuários finais do sistema, permitindo interação entre eles, visualização de informações, bem como realizar tarefas em comum. Esta interface é composta por três módulos básicos (Figura 5): Janela de Inspeção, Interface de Colaboração e Interface de Comunicação.



**Figura 5 – Interface Gráfica do Cliente Giga-Colab**

A Janela de Inspeção (Indicação 1) permite aos usuários do sistema visualizar informações e estado de entidades referentes a uma determinada Sala. Dentre essas informações destacam-se: nome da atividade, usuários, ferramentas associadas, bem como o repositório de conteúdo disponível para colaboração. A Janela de Inspeção também provê mecanismos de notificação de eventos, como por exemplo, mudança de estado (presente, ausente, etc.) de um determinado usuário da Atividade, ou até mesmo a indicação se um usuário está utilizando uma determinada aplicação colaborativa através do pedido de bloqueio.

A Interface de Colaboração (Indicação 2) dispõe de informações relevantes ao contexto geral do sistema, pois é através desta interface que os participantes apresentam seus documentos, vídeos e discussões de forma totalmente colaborativa. O usuário pode, por exemplo, visualizar vídeos sob demanda (Cliente JSC), utilizar o *whiteboard* ou uma aplicação de forma colaborativa com o uso do controle de concorrência. Tais informações são apresentadas em suas respectivas abas.

A Interface de Comunicação (Indicação 3) possibilita a configuração e visualização das funcionalidades de comunicação (áudio, vídeo e texto) de cada participante da colaboração (Cliente DCS). Atualmente, esta interface suporta até 16 (dezesseis) participantes simultâneos.

## 5. Considerações Finais

Este artigo apresentou a Giga-Colab, uma proposta de arquitetura de um ambiente virtual colaborativo. O objetivo foi reaproveitar uma solução já existente (jVE) e permitir a integração dinâmica de aplicações multimídia através da utilização de tecnologias de serviços *Web* e padrões abertos. Como forma de validação, a arquitetura foi instanciada para um protótipo que está em desenvolvimento no contexto do Projeto GIGA-RNP. Duas aplicações multimídia – vídeo sob demanda e videoconferência – foram utilizadas no sistema colaborativo e demonstraram a funcionalidade de integração.

O projeto da arquitetura trouxe os seguintes benefícios: (i) reutilização de uma infra-estrutura que facilite o desenvolvimento de aplicações colaborativas; (ii) fraco acoplamento na integração de aplicações externas ao ambiente colaborativo; e (iii) possibilidade de interoperabilidade com aplicações multimídia já existentes. Adicionalmente, também se pode citar como contribuição o emprego de funcionalidades avançadas de transmissão e recepção de vídeo para apoiar atividades colaborativas.

No contexto do projeto Giga-RNP, estão em andamento os seguintes trabalhos: (i) realização de experimentos para análise da solução na rede Giga; (ii) refinamento e adição de novas funcionalidades para o protótipo (por exemplo, vídeo ao vivo) e; (iii) desenvolvimento de serviços para o LIT/INPE (Laboratório de Integração e Testes do Instituto Nacional de Pesquisas Espaciais), de forma a possibilitar o uso de ferramentas colaborativas para acompanhamento de atividades de testes na construção de satélites.

Como trabalhos futuros, pode-se citar a possibilidade de uso de um modelo de política de colaboração para definir a semântica de integração entre os serviços [Gomes et al. 2006] e evolução da arquitetura para adição de novas características de um modelo orientado a serviços [Jorstad et al. 2005].

## Referências

- AccessGrid, “Access Grid”, Disponível em: <http://www.accessgrid.org>.
- Agora, “Agora – The Online Meeting Tool”, Lancaster University, Disponível em: <http://agora.lancs.ac.uk>.
- Alonso, G. et al. (2004), “Web Services: Concepts, Architectures and Applications”, Springer.
- Axis, A. Disponível em: <http://ws.apache.org/axis>
- Beavers, J. et al. (2004), “The Learning Experience Project: Enabling Collaborative Learning with ConferenceXP”, Technical Report MSR-TR-2004-42.
- Borghoff, U. M. e Schlichter, J. H. (2000), “Computer-Supported Cooperative Work: Introduction to Distributed Applications”, Springer, USA.
- Ellis, C. A., Gibbs, S. J. e Rein, G. L. (1991), “Groupware: some issues and experiences”, In *Communications of the ACM*, v.34, n., p.35-58.
- FFmpeg, Disponível em: <http://ffmpeg.mplayerhq.hu>.

- Fuchs, H. e Farber, N. (2005), "ISMA Interoperability and Conformance," In *IEEE Multimedia*, 2(2), p. 96-102.
- Gerosa, M. A. (2006), "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).
- Gomes, R. L., Hoyos-Rivera, G. J. H. e Courtiat, J. P. (2006), "Um Ambiente para Integração de Aplicações Colaborativas". In: Anais do Simpósio Brasileiro em Sistemas Colaborativos.
- Harada, C. Y. (2001), "Arquitetura para suporte a aplicações colaborativas", 317p. Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo, São Paulo.
- H.323, ITU Recommendation (1999), "Packet-base multimedia communications systems".
- Java Stream Assembly (2004), "API Programmer's Guide", Sun Microsystems Inc.
- Jorstad, I., Dustdar, S., Thanh, D. V. (2005) "A Service Oriented Architecture Framework for Collaborative Services", In: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, pp. 121-125.
- Kahler, H., Mørch, A., Stiemerling, O., Wulf, V. (2000) "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.
- Steinmetz, R. e Nahrstedt, K. (2004) "Multimedia Systems" X.Media.publishing, Springer.
- Licea, G. (2006), "Supporting reusability in fixed and mobile groupware applications", In *International Journal of Computers and Applications*, V.28(2), p. 99-111. ACTA Press.
- LiveMedia Library, Disponível em: <http://www.live555.com/liveMedia>
- Rosenberg, J. et al. (2002), "SIP: Session Initiation Protocol", RFC 3261, Internet Engineering Task Force, Disponível em: <http://www.ietf.org/rfc/rfc3261.txt>.
- Vasconcelos, M. A. V. M. e Filho, G. L. S. (2004) "Dynavideo Conference System, Um Sistema de Videoconferência H.323". In Anais do WebMedia/LA-Web Joint Conference.
- Voxgratia *Open-Source VoIP*, Disponível em: <http://www.voxgratia.org>
- Wu, W. et al. (2004) "Design and Implementation of a Collaboration Web-services system", In *Journal of Neural, Parallel & Scientific Computations*, Volume 12.

# Uma Infra-estrutura Colaborativa de Apoio ao Desenvolvimento Distribuído Baseado em Componentes

João Paulo F. de Oliveira, Talles Brito, Yuri Morais, Amílcar Soares,  
Adriana E. de Oliveira, Sebastião Rabelo Jr, Glêdson Elias

Departamento de Informática  
Universidade Federal da Paraíba (UFPB) – João Pessoa, PB – Brasil

{joaopaulo,talles,yuri,amilcar,drill,sebastiao,gledson}@compose.ufpb.br

**Abstract.** *Several organizations have adopted the distributed software development paradigm as a mean to reduce costs and development time, and also improve quality of products. As a consequence of geographic distribution of such organizations, tools and platforms are needed to improve collaboration and to control the software production process. In such a context, this paper presents a collaborative infrastructure for supporting component-based distributed development, which integrates facilities for access control, configuration management, version control and reuse metrics management.*

**Resumo.** *Diversas empresas estão passando a adotar o paradigma de desenvolvimento distribuído de software, com o objetivo de reduzir os custos e o tempo de desenvolvimento, bem como melhorar a qualidade dos seus produtos. Em decorrência da distribuição geográfica dessas empresas, ferramentas e plataformas são necessárias para melhorar a colaboração e controlar o processo de produção do software. Neste sentido, este artigo apresenta uma infra-estrutura colaborativa de apoio ao desenvolvimento distribuído baseado em componentes de software, que integra facilidades de controle de acesso, gerência de configuração, controle de versões e gerência de métricas de reuso.*

## 1. Introdução

Tem se tornado cada vez mais oneroso desenvolver software no mesmo espaço físico, na mesma organização ou até mesmo no mesmo país [1]. Assim, em um ambiente extremamente competitivo, empresas de Tecnologia da Informação (TI) que desejam continuar crescendo, ou até mesmo sobrevivendo, necessitam mudar inevitavelmente seus processos de desenvolvimento de software. No entanto, de acordo com Aoyama [2], os métodos tradicionais de desenvolvimento de software não permitem que tais empresas possam atingir os resultados esperados.

Neste sentido, na engenharia de software, o Desenvolvimento Distribuído de Software (DDS) e o Desenvolvimento Baseado em Componentes (DBC) destacam-se como abordagens que têm o potencial de tornar realidade a redução da complexidade, do tempo e do custo de desenvolvimento, bem como melhorar a qualidade do software produzido [3] e a possibilidade de obter recursos em âmbito global [4].

Vale ressaltar que existe uma relação entre as abordagens de DBC e DDS. Esta relação explicitada na forma de um mercado de software, como definido por Wallnau [5], onde o produtor fornece componentes de software para diversos consumidores e um consumidor adquire componentes de diversos produtores, ambos (produtores e consumidores) dispersos geograficamente. É importante destacar que, nesta relação, consumidores desempenham o papel de integradores de software, desenvolvendo novos

componentes a partir do reuso e composição de componentes previamente existentes e desenvolvidos de forma independente por terceiros.

Entretanto, tanto o DBC quanto o DDS apresentam alguns fatores limitantes. No DBC, para Crnkovic [6], o desafio está na atual dificuldade de identificar, selecionar, negociar e recuperar componentes que atendam aos requisitos especificados, e, além disso, que possuam alto grau de reusabilidade e qualidade. Já no DDS, o principal desafio é inerente à própria característica da distribuição que dificulta a colaboração e a cooperação entre departamentos e pequenos grupos de desenvolvedores que trabalham em conjunto, mas estão localizados em cidades ou países diferentes [7].

Para Tommarello [8], trabalho com equipes distribuídas é sempre um desafio. No entanto, esse desafio pode ser reduzido com o uso de ferramentas e ambientes de *groupware* para coordenação e comunicação das equipes durante um processo de desenvolvimento distribuído de software.

Neste contexto, com o objetivo de superar as limitações expostas tanto para o DBC quanto para o DDS, este artigo apresenta a implementação de um serviço de repositório compartilhado e distribuído, que provê facilidades de *groupware* atuando como uma infra-estrutura colaborativa de apoio ao desenvolvimento distribuído de componentes de software. O serviço de repositório provê facilidades e suporte a modelos de negócios, certificação de componentes, aspectos de segurança (autenticação, visibilidade e controle de acesso), esquema de nomeação, controle de versões e informações de reuso.

O serviço de repositório proposto é dito ser compartilhado, pois vários produtores podem registrar seus artefatos. Por outro lado, o serviço de repositório é dito distribuído porque uma implementação aderente pode ser baseada em um conjunto de entidades geograficamente dispersas que cooperam para prover as funcionalidades do serviço de repositório.

Vale ressaltar que o serviço de repositório é projetado de forma a manipular artefatos de software (*assets*) em geral, que incluem quaisquer artefatos produzidos ou reusados no DBC. Exemplos de artefatos são: códigos executáveis, códigos fonte, especificações de interfaces, documentações, casos de uso, diagramas UML e planos de teste.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2, o serviço de repositório é comparado com alguns trabalhos relacionados. A Seção 3 introduz o *framework* arquitetural *ComponentForge* [9], no qual o serviço de repositório representa a infra-estrutura colaborativa de apoio ao desenvolvimento distribuído de componentes de software. A Seção 4 apresenta as principais características e funcionalidades do serviço de repositório que provêm apoio a colaboração e cooperação. Em seguida, a Seção 5 apresenta o projeto arquitetural do serviço de repositório e sua implementação. Por fim, a Seção 6 apresenta algumas considerações finais.

## **2. Trabalhos Relacionados**

Avaliando as ferramentas e plataformas de DBC, disponíveis na academia e indústria, torna-se evidente a ausência de um ferramental de *groupware* para coordenação e comunicação das equipes durante um processo de desenvolvimento distribuído de componentes de software. Para evidenciar esta ausência de ferramental de *groupware* no contexto de DBC, esta seção apresenta uma breve comparação das funcionalidades providas pelo serviço de repositório proposto neste artigo com os

principais repositórios disponíveis na literatura e mercado, tais como: *SPARS-J* [10], *OSCAR* [11], *CodeBroker* [12] e *ComponentSource*[13].

A fim de facilitar a análise dos repositórios, resolvemos adotar o Modelo *Clover* [14] para classificação do apoio computacional à colaboração. Neste modelo, uma plataforma de *groupware* deve fornecer três classes de serviços complementares: *coordenação*, *produção* e *comunicação*.

As facilidades de *coordenação* estão relacionadas com os papéis que cada usuário pode exercer e como essas atividades são coordenadas de forma a alcançar o objetivo final. O *ComponentSource*, o *SPARS-J* e o *CodeBroker* adotam mecanismos para a coordenação de atividades bastante simples, pois não permitem aos usuários exercerem diferentes papéis com distintas responsabilidades. Em ambos os casos, os produtores apenas registram seus componentes implementados e prontos, não tendo a noção de equipes distribuídas que cooperam durante o processo de desenvolvimento de software.

De forma similar ao *ComponentSource*, *SPARS-J* e *CodeBroker*, o *OSCAR* também não adota mecanismos sofisticados para a coordenação de atividades. Na literatura, apenas é possível identificar que o *OSCAR* adota um simples esquema de histórico (*log*), que armazena quem acessou, qual artefato foi acessado e qual foi o propósito do acesso. No entanto, estas informações são apenas usadas para auditoria, mas não podem ser usadas como mecanismo de controle na definição de diferentes papéis aos usuários.

Ao contrário das principais propostas identificadas na literatura e indústria, o serviço de repositório proposto neste artigo adota mecanismos sofisticados para a coordenação das atividades, permitindo aos usuários exercerem diferentes papéis com distintas responsabilidades. O serviço de repositório adota o modelo de controle de acesso denominado RBAC (*Role-Based Access Control*) [15], onde os usuários são classificados em diferentes papéis e diferentes permissões de acesso são associadas a cada papel desempenhado pelo usuário. Deste modo, de forma simples e de fácil manutenção, é possível controlar que operações as equipes distribuídas de desenvolvedores podem executar durante o processo de desenvolvimento de componentes de software.

De acordo com Laurillau [14], a segunda classe de serviço requerido por uma plataforma de *groupware* está relacionada com facilidades de *produção*. Neste contexto, a produção está relacionada com os resultados das atividades desenvolvidas em grupo ou aos dados compartilhados por múltiplos usuários. No caso específico de DBC e DDS, a produção está diretamente relacionada às facilidades de gerência de configuração, controle de versão e certificação dos componentes de software. Além disso, a produção também diz respeito à quantidade de produtores e consumidores que podem fazer uso das facilidades da plataforma de *groupware*, como também à quantidade de componentes que podem ser manipulados.

Em relação às facilidades de produção, o *CodeBroker*, o *SPARS-J* e o *OSCAR* adotam uma base de dados de escala restrita a uma determinada instituição, que impede a escalabilidade do serviço oferecido, tanto em termos da quantidade de produtores e consumidores que usam os serviços, quanto na quantidade de componentes manipulados. Embora o *ComponentSource* atue provendo componentes de software comerciais (COTS – *Commercial Off-The-Shelf*) em escala global e multi-institucional, o aspecto centralizado de sua arquitetura o torna não escalável em termos da quantidade

de produtores e consumidores que usam os serviços, quanto na quantidade de componentes manipulados.

Além disso, o *CodeBroker* e o *SPARS-J* também não adotam qualquer mecanismo de gerência de configuração, controle de versão e certificação dos componentes. Logo, no *CodeBroker* e *SPARS-J*, as facilidades de produção são bastante limitadas, abrangendo apenas a possibilidade de registrar e recuperar artefatos. Considerando a gerência de configuração e o controle de versões suportados pelo OSCAR, pode-se perceber que o mesmo adota abordagens baseadas na integração de soluções de terceiros e centradas apenas no controle de código fonte, tais como CVS. Além disso, o OSCAR não trata aspectos relacionados à certificação de componentes.

Por outro lado, no que se refere às facilidades de produção, o serviço de repositório proposto neste artigo define mecanismos próprios de gerência de configuração e controle de versões, capazes de controlar de forma customizada a evolução dos diferentes tipos de artefatos, ou seja, não apenas código fonte. Além disso, o serviço de repositório suporta diferentes estratégias de certificação de componentes, cujos certificados podem ser registrados por diferentes entidades certificadoras que adotam variados modelos de certificação. Por fim, a natureza compartilhada e distribuída do serviço de repositório torna a solução escalável em termos do número de produtores e consumidores participantes, como também da quantidade de componentes manipulados.

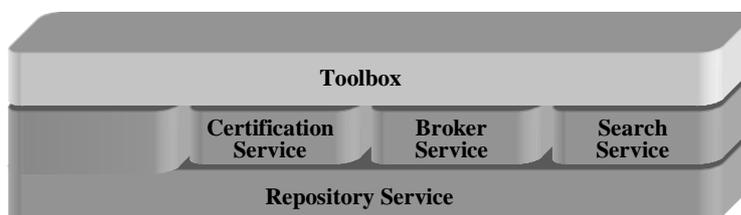
Por fim, ainda de acordo com Laurillau [14], a terceira classe de serviço requerido por uma plataforma de *groupware* está relacionada com facilidades de *comunicação*, que trata das relações diretas entre os usuários, como por exemplo, uma possível troca de mensagens entre desenvolvedores e consumidores. O *ComponentSource*, por ser um repositório comercial, oferece *chat* e fórum para que os usuários possam obter mais detalhes sobre os artefatos que desejam adquirir. O *OSCAR*, *SPARS-J* e *CodeBroker* não oferecem nenhum apoio a facilidades de comunicação entre produtores e consumidores.

Diferentemente, em relação às facilidades de comunicação, o serviço de repositório apresentado neste artigo oferece um conjunto de informações de métricas de reuso associadas aos artefatos registrados. Estas informações contêm os comentários dos consumidores sobre suas experiências de instalação, configuração e uso de um determinado artefato, como também as respostas apresentadas pelos desenvolvedores a tais comentários. Neste sentido, as informações de reuso permitem os produtores e consumidores compartilharem informações que facilitam a manutenção dos artefatos e também auxiliam os consumidores na seleção dos mesmos.

### 3. ComponentForge

Considerando as limitações das propostas atualmente disponíveis, este artigo descreve as facilidades de *groupware* suportadas pela implementação de um serviço de repositório compartilhado e distribuído. Este serviço de repositório proposto atua como uma infra-estrutura colaborativa de apoio ao desenvolvimento distribuído de componentes de software. Atuando como um elemento de infra-estrutura, o serviço de repositório está inserido no contexto de um *framework* arquitetural de alto nível, denominado *ComponentForge*, que provê apoio a processos de desenvolvimento baseados em reuso de componentes. Esta sessão tem por objetivo descrever sucintamente os elementos que compõem o *framework*, contextualizando assim o papel do serviço de repositório.

Como ilustrado na Figura 1, o *ComponentForge* adota uma abordagem de arquitetura orientada a serviços (SOA – *Service-Oriented Architecture*), na qual um conjunto de serviços distribuídos, compartilhados, independentes e fracamente acoplados comunicam-se e colaboram entre si através de interfaces e protocolos de comunicação bem definidos.



**Figura 1. Framework Arquitetural do ComponentForge**

Coletivamente, o conjunto de serviços do *framework* provê facilidades para armazenar, indexar, buscar, recuperar, certificar e negociar variados artefatos de software (*assets*). Além disso, o *ComponentForge* inclui facilidades de gerência de configuração, gerência de métricas de reuso, como também aspectos de segurança relacionados à autenticação e controle de acesso.

O serviço de repositório (*repository service*) é responsável por armazenar, localizar, recuperar e gerenciar *assets*. O serviço de repositório é suportado por uma coleção de entidades cooperantes e distribuídas denominadas *containers*, que, conjuntamente, provêm facilidades às demais entidades do *framework* arquitetural. Apesar de não estar explícito na Figura 1, o serviço de repositório pode se comunicar com uma ou mais instâncias de *toolbox*, *certification service*, *broker service* e *search service*.

A certificação de componentes é uma atividade que procura validar as funcionalidades e os níveis de qualidade dos componentes [16]. O *ComponentForge* foi concebido de forma a permitir a coexistência de diversos serviços de certificação (*certification service*), possibilitando a existência de múltiplos e complementares processos de certificação. Estes serviços são responsáveis por certificar não apenas componentes individuais, mas também as práticas e os processos adotados pelos produtores.

Da mesma forma como acontece na produção de softwares em geral, produtores de componentes reusáveis também podem estar interessados ou não em ter um retorno financeiro. Logo, podem-se identificar dois tipos de componentes: os *componentes sem modelo de negócio*, que podem ser distribuídos para qualquer consumidor sem restrições, e os *componentes com modelo de negócio*, cujas formas de negociação são regidas por contratos especificados nos modelos de negócios, definidos por seus respectivos produtores. Portanto, componentes podem ser adquiridos sob uma variedade de modelos de negócios.

Sendo assim, o serviço de negociação (*broker service*) tem a função de prover mecanismos para assegurar que um componente será entregue em conformidade com os modelos de negócios especificados pelo produtor. Depois que um consumidor atender todas as condições dos modelos de negócios de um componente, o serviço de negociação pode acessar o serviço de repositório para recuperar o componente e entregá-lo ao consumidor. Os produtores de componentes podem desenvolver seus próprios serviços de negociação ou indicar serviços já existentes para tratar os modelos de negócios sob os quais seus componentes podem ser negociados.

Vale ressaltar que, mesmo componentes desenvolvidos sob uma licença de software livre, também podem ter que satisfazer restrições especificadas em um determinado modelo de negócio. Neste caso, tais componentes somente podem ser adquiridos através de um determinado serviço de negociação. Nesta abordagem, por exemplo, produtores de software livre podem manter, em seus respectivos serviços de negociação, catálogos de consumidores aptos a recuperar seus componentes ou consumidores que já realizaram a recuperação de seus componentes.

Para permitir a busca e a recuperação de componentes armazenados, o *ComponentForge* define o serviço de busca (*search service*). Este serviço consulta o serviço de repositório e recupera metadados para realizar a indexação. O *ComponentForge* explora um modelo de representação de componentes, denominado X-ARM [17]. Tal abordagem torna a arquitetura mais flexível, pois permite que diferentes serviços de busca adotem diferentes algoritmos de indexação e linguagens de consulta. Desta forma, um serviço de busca pode indexar componentes de um determinado domínio de aplicação específico, e, além disso, prover uma linguagem de consulta especializada para tal domínio. Por outro lado, outro serviço de busca pode optar por indexar componentes de forma independente de domínio de aplicação.

Por fim, o conjunto de ferramentas (*toolbox*) permite a interação dos usuários com os demais serviços do *ComponentForge*. Vale ressaltar que o baixo acoplamento entre os serviços que compõem o *framework* facilita a configuração, reuso, manutenção, evolução e extensão, todos considerados importantes propriedades arquiteturais de aplicações distribuídas, conforme mencionado por [18].

#### **4. O Serviço de Repositório como uma Infra-estrutura Colaborativa**

Colaboração é uma maneira de desenvolver atividades em grupo. Visando a colaboração no desenvolvimento de componentes de software, nesta seção serão apresentadas as principais funcionalidades do serviço de repositório, evidenciando aquelas diretamente relacionadas às atividades colaborativas.

Para a classificação das funcionalidades do serviço de repositório utilizaremos novamente o Modelo *Clover* [14]. Como mencionado, de acordo com este modelo, uma plataforma de *groupware* deve prover três classes de serviços: *coordenação*, *produção* e *comunicação*.

Nas facilidades de coordenação, a estruturação de uma equipe inclui a definição dos papéis dos usuários, da hierarquia, dos subgrupos e das permissões dos participantes[19]. A fim de gerenciar quais usuários terão permissão para executar determinadas operações e quais serviços terão permissões de acesso aos artefatos, o serviço de repositório possui um sistema de controle de acesso. Além disso, o serviço de repositório provê um esquema de visibilidade para que o desenvolvedor possa restringir o acesso a seus artefatos.

No trabalho em grupo, as facilidades de produção estão relacionadas à operação conjunta dos participantes no espaço compartilhado, visando a realização de tarefas [19]. Além disso, as facilidades de produção devem permitir que os desenvolvedores possam manipular, refinar e organizar os artefatos produzidos. Assim, no serviço de repositório, o controle de versões tem por objetivo permitir que diversos desenvolvedores possam colaborar na evolução de um mesmo artefato de forma organizada e sem conflitos. Além disso, o esquema de nomeação proporciona a recuperação não ambígua de artefatos, facilitando assim a organização da produção de todo o conteúdo do serviço de repositório.

Através das facilidades de comunicação, o grupo debate pontos de vista para alinhar e refinar as idéias [19]. Neste contexto, o serviço de repositório define o que chamamos de informações de reuso, onde os consumidores podem expor suas avaliações a respeito dos artefatos recuperados.

Neste ponto, apenas identificamos as principais funcionalidades do serviço de repositório que estão diretamente relacionadas às atividades colaborativas. A seguir, as próximas subseções detalham os conceitos e funcionalidades sobre nomeação, controle de acesso aos usuários e demais serviços do *framework*, visibilidade dos artefatos, controle de versões dos artefatos e informações de reuso.

#### 4.1 Esquema de Nomeação

O serviço de repositório é composto por um conjunto de entidades cooperantes e distribuídas denominadas *containers*. A Figura 2 ilustra a visão distribuída e compartilhada do serviço de repositório, em que o mesmo é uma entidade representativa do conjunto de facilidades providas pelos *containers*.

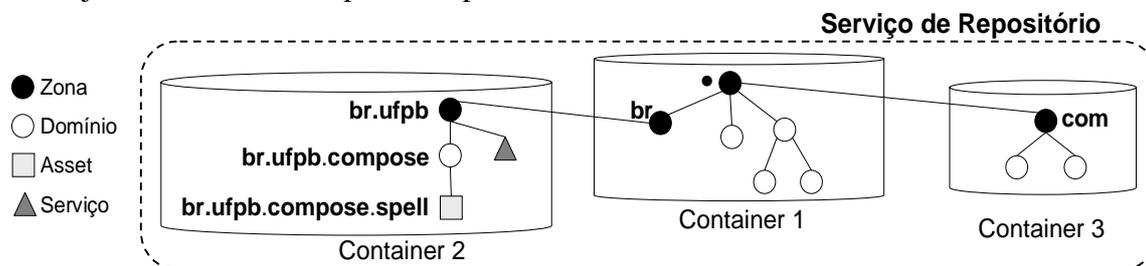


Figura 2. Árvore de Nomes Hierárquicos

Assim, considerando a natureza distribuída e compartilhada do serviço de repositório, as atividades de *groupware* relacionadas às facilidades de produção dos artefatos armazenados no serviço de repositório utilizam um esquema que permite a recuperação não ambígua de artefatos. Em tal esquema, os nomes são organizados em uma árvore hierárquica onde as folhas são os artefatos armazenados e serviços oferecidos. Nesta árvore, os nós internos correspondem a dois tipos de entidades (zonas e domínios).

Zonas representam produtores de artefatos e suas famílias de produtos e serviços. A fim de reduzir o esforço de coordenação e possibilitar uma melhor organização dos artefatos, uma zona também pode ser subdividida em uma estrutura hierárquica, na qual os nós são denominados domínios, cuja função é agrupar um conjunto de artefatos para um determinado domínio de aplicação.

Na composição dos identificadores hierárquicos, os nomes de zonas, domínios, serviços e artefatos são separados por pontos. Assim, a Figura 2 ilustra uma árvore de nomes hierárquicos onde *br.ufpb* refere-se à zona da UFPB, que possui o domínio *br.ufpb.compose*, que representa o grupo COMPOSE e este possui um artefato denominado *br.ufpb.compose.spell*. Vale ressaltar que a zona raiz do serviço de repositório é identificada apenas por um ponto.

Além de definir identificadores globalmente únicos, o esquema hierárquico proposto também provê transparência de localização, pois os nomes não referenciam a localização física, mas apenas a organização lógica de produtores, artefatos e serviços.

#### 4.2 Controle de Acesso

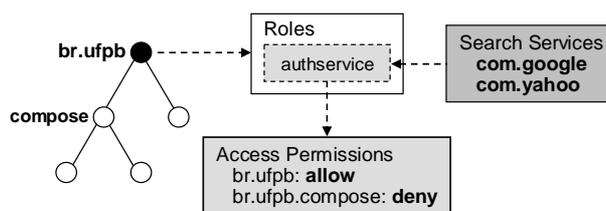
O controle de acesso provido pelo serviço de repositório é um mecanismo relacionado às atividades de *groupware* associadas às facilidades de coordenação. O controle de

acesso gerencia as operações que podem ser executadas por um usuário, e, também é o responsável pela definição das permissões de acesso dos serviços aos artefatos, zonas e domínios.

Os usuários do serviço de repositório podem executar tarefas distintas. O gerente de container realiza tarefas administrativas, registrando zonas autorizadas e verificando dados estatísticos relacionados com o container. Um administrador de zona gerencia uma determinada zona, criando e removendo domínios, assim como registrando, removendo e atualizando informações sobre desenvolvedores autorizados. Um desenvolvedor é responsável por registrar, atualizar e remover artefatos em sua respectiva zona. Já um consumidor pode procurar e adquirir artefatos previamente armazenados. Além disso, os serviços de certificação, negociação e busca também podem acessar as facilidades providas pelo serviço de repositório.

No contexto de controle de acesso, o serviço de repositório adota um mecanismo semelhante ao modelo RBAC (*Role-Based Access Control*) [15], onde as permissões de acesso são associadas a papéis, que, por sua vez, são atribuídos aos usuários e serviços. Assim, quando um novo usuário ou serviço é cadastrado, o mesmo é associado a um conjunto de papéis, e, assim, suas permissões de acesso são automaticamente atribuídas. Conseqüentemente, o gerenciamento do controle de acesso torna-se mais simples e viável.

Permissões de acesso associadas para usuários e serviços são diferentes. No caso dos usuários, permissões de acesso simplesmente referem-se para operações permitidas. Entretanto, no caso de serviços, permissões de acesso referem-se a zonas e domínios permitidos ou bloqueados. A Figura 3 exemplifica tais permissões para serviços. A zona *br.ufpb* tem o papel *authservice*, composto de um conjunto de permissões de acesso que controlam quais componentes podem ser acessados pelos serviços de busca *com.google* e *com.yahoo*. Neste caso, é importante enfatizar que os serviços *com.google* e *com.yahoo* também devem estar registrados no serviço de repositório.



**Figura 3. Permissões de Acesso para Serviços**

Nas permissões de acesso, é possível permitir ou bloquear acesso para componentes específicos ou todos os componentes dos domínios indicados de uma maneira recursiva. Na Figura 3, veja que os serviços de busca *com.google* e *com.yahoo* têm permissão para acessar todos os artefatos da zona *br.ufpb*, recursivamente incluindo todos os seus domínios abaixo, mas recursivamente excluindo o domínio *br.ufpb.compose* e todos abaixo dele.

### 4.3 Visibilidade Externa

A fim de facilitar o desenvolvimento colaborativo e controlar o acesso aos artefatos por diferentes equipes distribuídas de desenvolvedores, o serviço de repositório explora o conceito de visibilidade. Assim, para cada artefato registrado em uma zona ou domínio, podem ser definidos esquemas que indicam as outras zonas ou domínios cujas equipes

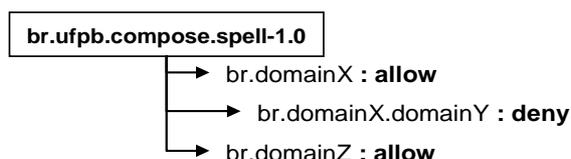
de desenvolvedores são autorizadas ou proibidas de recuperar o artefato. Vale ressaltar que este esquema só pode ser modificado pelos desenvolvedores que o configuraram.

A visibilidade é considerada um mecanismo relacionado com as atividades de *groupware* associadas às facilidades de coordenação, pois proporciona um melhor controle dos artefatos de uma zona ou domínio, evitando que artefatos não concluídos sejam recuperados por outros desenvolvedores não autorizados e até mesmo por consumidores.

Quando um desenvolvedor autorizar ou proibir uma determinada zona ou domínio, esta autorização ou proibição é recursivamente aplicada a todos os domínios filhos em qualquer nível de profundidade, exceto quando esta autorização ou proibição é explicitamente negada para algum destes subdomínios.

Vale ressaltar que se um artefato não possui um esquema de visibilidade configurado, qualquer consumidor pode recuperar este artefato. Neste caso, apenas desenvolvedores permitidos no domínio onde o artefato está registrado podem executar operações envolvendo a manipulação das informações do artefato.

A Figura 4 ilustra o uso do esquema de visibilidade. Neste caso, o artefato *br.ufpb.compose.spell-1.0* pode ser acessado por todos os desenvolvedores dos domínios *br.domainX* e *br.domainZ*, incluindo aqueles dos seus respectivos subdomínios de qualquer nível de profundidade, exceto do subdomínio *br.domainX.domainY*.



**Figura 4. Esquema de visibilidade**

#### 4.4 Controle de Versões

No serviço de repositório, diversos desenvolvedores podem estar manipulando um mesmo artefato. Desta forma, um mecanismo para controle de versões dos artefatos é fundamental.

O serviço de repositório oferece este tipo de apoio durante o processo de desenvolvimento dos artefatos, permitindo assim, que os desenvolvedores pertencentes a um mesmo grupo de trabalho e distribuídos geograficamente, participem da produção de um mesmo artefato simultaneamente. Cada desenvolvedor pode trabalhar em uma versão base e submeter a sua versão modificada, sem correr o risco de sobrescrever a versão submetida por outros desenvolvedores. Para tal, o serviço de repositório oferece um mecanismo de bloqueio de artefatos.

Quando o desenvolvedor recupera um artefato que servirá como versão base para o desenvolvimento, o serviço de repositório bloqueia este artefato para que outros desenvolvedores não possam modificá-lo. Sendo assim, este artefato somente poderá ser atualizado por outros desenvolvedores depois que o mesmo for desbloqueado pelo desenvolvedor que o bloqueou. Assim, o registro de uma nova versão de um artefato automaticamente desbloqueia a versão base deste artefato. Este esquema de proteção garante o controle de versões para os artefatos que estão em desenvolvimento no serviço de repositório. O controle de versões adotado é semelhante aos empregados em outros sistemas de controle de versão comerciais, diferenciando pelo fato de que artefatos não são somente código, mas qualquer elemento produzido durante o processo de desenvolvimento.

No serviço de repositório, é possível que um conjunto de desenvolvedores trabalhem cooperativamente em uma versão de um artefato sem que seja preciso torná-la pública aos consumidores. Para tal, os artefatos são marcados como *registrados* ou *publicados*. Um artefato no estado registrado somente pode ser recuperado pelos desenvolvedores e serviços de busca e certificação autorizados. Por outro lado, um artefato no estado publicado pode ser recuperado pelos desenvolvedores e todos os serviços autorizados (serviços de busca, certificação e negociação), como também pelos consumidores. Para Gerosa [19], este tipo de mecanismo é um importante aspecto que deve ser disponibilizado por plataformas de *groupware*.

Vale ressaltar que o controle de versões provido pelo serviço de repositório também permite identificar e recuperar um histórico de evolução dos artefatos. Com isto, edições anteriores podem ser facilmente recuperadas, possibilitando ao usuário do serviço a comparação de versões.

#### **4.5 Informações de Reuso**

Estudos demonstram que a prática de reutilização ainda é muito dependente de comunicação entre as equipes produtoras e consumidoras de componentes. A documentação é suficiente para descrever o conhecimento de experiências, mas a comunicação direta é necessária para resolver diversos problemas e estabelecer relações de confiança entre os produtores e consumidores de software [20].

Então, para o desenvolvedor de artefatos é importante receber um retorno dos consumidores o mais cedo possível. Além disso, a possibilidade de o consumidor opinar sobre um determinado artefato, pode fazer com que o mesmo se sinta parte do processo de desenvolvimento do artefato [21].

Com a finalidade de estabelecer um canal de comunicação entre consumidores e desenvolvedores, o serviço de repositório define o que chamamos de informações de reuso. Estas consistem de um conjunto de informações referentes ao grau de satisfação de consumidores que já adquiriram um determinado artefato, seus comentários sobre a aquisição, área de aplicação do artefato, além de respostas emitidas pelos desenvolvedores do artefato sobre os comentários do consumidor. Neste contexto, as informações de reuso consistem em um elemento facilitador da comunicação e da cooperação entre usuários e desenvolvedores de maneira a aumentar a qualidade dos artefatos desenvolvidos.

É importante ressaltar que estas informações de reuso ficam a disposição de futuros consumidores, que podem se valer da experiência relatada por outros consumidores para ajudar no processo de escolha de um determinado artefato.

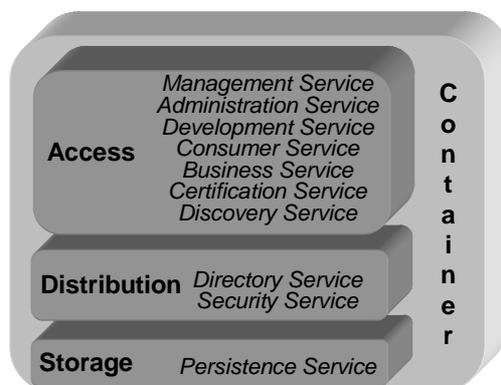
### **5. Projeto e Implementação do Serviço de Repositório**

Em conjunto, as interfaces do serviço de repositório definem mais de 800 operações, que, atualmente, estão implementadas no protótipo piloto. No entanto, devido à limitação de espaço, as interfaces e o projeto do protótipo não serão descritos. A seguir, descreveremos a arquitetura definida para o repositório juntamente com a função e delimitação de cada um de seus serviços. Por fim, uma visão geral sobre a implementação desta arquitetura será apresentada.

#### **5.1 Projeto Arquitetural**

Como mencionado anteriormente, as facilidades do serviço de repositório são providas pelos *containers*, que possuem um projeto arquitetural no estilo em camadas, onde cada

uma destas provê um conjunto de componentes, que também adotam a abordagem de serviços. A Figura 5 ilustra a arquitetura em camadas do *container*.



**Figura 5. Camadas e serviços do container**

A camada de acesso (*access layer*) permite a interação das ferramentas e dos outros serviços do *ComponentForge* com o container e com o serviço de repositório como um todo. Nesta camada, o serviço de gerenciamento (*management service*) oferece um conjunto de interfaces que permite a administração do container, tais como o registro, ajuste e remoção de zonas. O serviço de administração (*administration service*) provê operações para o gerenciamento de zonas e domínios. O serviço do desenvolvedor (*development service*) provê operações para o registro, atualização, remoção e gerenciamento de artefatos. O serviço do consumidor (*consumer service*) provê mecanismos para recuperação de artefatos que não adotam modelos de negócios, enquanto que, o serviço de negociação (*business service*) permite a recuperação de artefatos que adotam modelos de negócios. O serviço de certificação (*certification service*) trata aspectos da certificação de artefatos. Por fim, o serviço de descoberta (*discovery service*) permite a recuperação de artefatos para posterior indexação em mecanismos de busca.

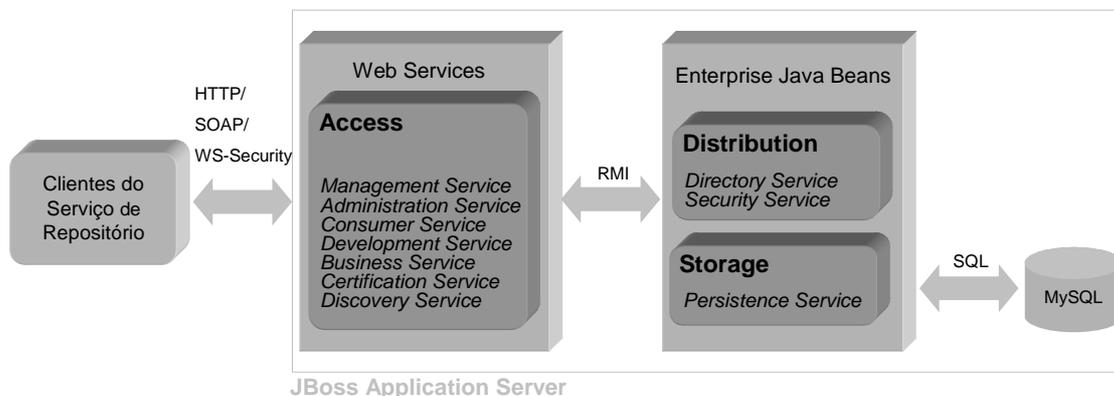
A camada de distribuição (*distribution layer*) trata de aspectos não funcionais do serviço de repositório, resolvendo questões de distribuição e segurança, tornando possível a descoberta transparente e a recuperação segura de artefatos armazenados. A fim de prover tais facilidades, a camada de distribuição oferece o serviço de diretório (*directory service*) e o serviço de segurança (*security service*).

Os *containers* também são responsáveis pelo armazenamento físico dos artefatos, formando uma base de dados distribuída. Dessa forma, cada *container* possui a camada de armazenamento (*storage layer*), que disponibiliza o serviço de persistência (*persistence service*). Este serviço é acessado pelos serviços da camada de acesso e de distribuição, oferecendo meios para armazenar, atualizar e recuperar artefatos.

## 5.2 Implementação

A Figura 6 apresenta uma visão geral da implementação do serviço de repositório, identificando as diversas tecnologias envolvidas. Devido ao grande volume de dados manipulados pelo serviço de repositório, surgiu a necessidade de uma plataforma que proporcione manutenibilidade, confiabilidade e escalabilidade ao sistema. A especificação *Java Platform Enterprise Edition* (Java EE) [22] consegue satisfazer estes requisitos, pois tem como principal objetivo o desenvolvimento em n-camadas. Para execução de um sistema Java EE é necessário um servidor de aplicações, e para isso foi selecionado o *JBoss Application Server* [23]. Este preenche os requisitos de qualquer

servidor de aplicações poderoso, como *clustering*, tolerância a falhas, balanceamento de cargas, *caching* e está de acordo com as mais recentes especificações do *Enterprise JavaBeans* (EJB).



**Figura 6. Visão geral sobre a implementação do serviço de repositório.**

Um dos requisitos chaves para sistemas colaborativos é que estes sejam compatíveis e adaptáveis aos variados ambientes de seus usuários [24]. Em função disso, e, pelo fato do *ComponentForge* ser baseado em uma arquitetura orientada a serviços (SOA), decidimos disponibilizar seus serviços via *Web Services* [25]. Esta decisão proporciona interoperabilidade, permitindo que instâncias implementadas em diferentes plataformas e linguagens de programação possam se comunicar simplesmente expondo suas interfaces. Além disso, *Web Services* favorece o acesso de diferentes maneiras, que podem incluir, por exemplo, interfaces do usuário baseadas em navegadores *Web*, aplicações *desktop* e dispositivos móveis.

Com o uso de *Web Services*, questões sobre segurança, integridade e confidencialidade constituem fatores críticos nas trocas de mensagens. Assim, se faz necessário adotar meios para tratar estas questões, que incluem autenticação, autorização e criptografia [26]. Para preencher esta lacuna, a OASIS (*Organization for the Advancement of Structured Information Standards*) desenvolveu a especificação WS-Security [27]. Este padrão define uma forma de tratar os vários cenários de segurança na comunicação baseada em mensagens SOAP. WS-Security define um único modelo que abstrai serviços de segurança, separando as características funcionais de segurança do sistema de sua implementação específica [26]. Para realizar o controle de acesso (Sessão 4.2), o serviço de repositório deve suportar mecanismos de autenticação dos usuários e integridade das informações. Para tal, a implementação adota as facilidades providas pelo WS-Security.

Por fim, em função da natureza compartilhada e distribuída do *ComponentForge*, mecanismos para o gerenciamento de dados persistentes e não persistentes do serviço de repositório são de fundamental importância. Assim, devido ao serviço de repositório adotar um esquema de *cache* de resolução de nomes, alguns dados devem permanecer em memória principal em vez de serem armazenados em memória persistente, proporcionando um melhor desempenho na resolução de nomes. Toda a questão de sincronização e transações destes dados é gerenciada pelo próprio *JBoss Application Server*, de maneira que ganhamos em produtividade ao eliminar qualquer tipo de programação propensa a erros.

Além disto, o ComponentForge manipula e armazena uma grande quantidade de dados persistentes. Assim, para oferecer suporte a acessos simultâneos e em larga escala, fez-se necessária a utilização de um SGBD robusto, rápido, multi-threaded e multi-usuário. Com isso, dentre os principais SGBDs hoje existentes, o MySQL (5.0) atendeu a esses requisitos [28], além de dar suporte às necessidades do projeto quanto às questões de armazenamento e acesso concorrente.

## 6. Considerações Finais

O Desenvolvimento Distribuído de Software, assim como o Desenvolvimento Baseado em Componentes, vêm ganhando cada vez mais importância na Engenharia de Software em função da qualidade dos produtos gerados quando são adotados processos, técnicas e ferramentas aderentes a tais abordagens de desenvolvimento. A interseção entre essas duas abordagens oferece um vasto campo para pesquisa, visto a necessidade cada vez mais latente de ferramentas que dêem apoio a ambas abordagens.

Conseqüentemente, a principal contribuição deste artigo é propor um serviço de repositório que define uma infra-estrutura colaborativa que oferece suporte à coordenação, comunicação e produção de componentes de software. Por exemplo, o serviço de repositório provê facilidades de controle de versão, autenticação, controle de acesso e visibilidade, que são fundamentais para o apoio à colaboração entre equipes distribuídas de desenvolvimento.

Além disso, em função da componentização, o serviço de repositório define uma arquitetura que proporciona manutenibilidade, reuso, composição, extensibilidade, integração e escalabilidade. Segundo Greenberg [29], o uso de componentes que encapsulam as complexidades do desenvolvimento de facilidades de *groupware* é uma maneira de impulsionar o desenvolvimento desta tecnologia.

Vale ressaltar que o compartilhamento e a distribuição providos pelo serviço de repositório, em conjunto, têm o potencial de promover o aumento da quantidade de componentes disponíveis no repositório. Conforme mencionado por Frakes[30], a disponibilidade de uma grande quantidade de componentes tem o potencial de incrementar o reuso de componentes.

Embora a implementação do serviço de repositório já contemple importantes requisitos funcionais, outras facilidades devem ainda serem incluídas em versões futuras, tais como histórico de ações (*logs*), sistema de mensagens síncronas (*chat*), apoio ao processo de gerenciamento de projetos, bem como a especificação e implementação dos demais serviços que compõem o *ComponentForge*.

## 7. Referências

- [1] Prikładnicki, R.; Marczak, S.; and Audy, J. L. (2006). "MuNDDoS: A Research Group on Global Software Development". In International Conference on Global Software Engineering (ICGSE'06)
- [2] Aoyama, M. (1998). "New age of software development: How component-based software engineering changes the way of software development". In International Workshop on Component-Based Software Engineering(CBSE'98)..
- [3] Szyperski, Clemens (2002). "Component Software: Beyond Object-Oriented Programming". Second Edition, Addison-Wesley.
- [4] Herbsleb, J.D., Grinter, R.(1999) "Splitting the organization and integrating the code: Conway's Law revisited", In 21<sup>th</sup> International Conference on Software Engineering (ICSE'99).

- [5] Wallnau, K. C.; Hissam, S. A.; Seacord, R. C.,(2001), “Building Systems from Commercial Components”, SEI Series in Software Engineering, Addison-Wesley
- [6] Crnkovic, Ivica. (2003) “Component-Based Software Engineering–New Challenges in Software Development”. In Information Technology Interfaces (ITI’03).
- [7] Kiel, L. (2003).“Experiences in Distributed Development: A Case Study”, In: Workshop on Global Software Development at ICSE 2003”, Oregon, EUA.
- [8] Tommarello, J. D.; Deek, Fadi P. (2002). “Collaborative Software Development: A Discussion of Problem Solving Models and Groupware Technologies”. In 35th Annual Hawaii International Conference on System Sciences (HICSS’02)
- [9] Oliveira, João Paulo F.; Santos, Michael S.; Elias, Gledson. (2006). “ComponentForge:Um Framework Arquitetural para Desenvolvimento Distribuído Baseado em Componentes. VI WDBC. Recife – PE.
- [10] Inoue, K.; *et al.* (2003) “Component Rank: Relative Significance Rank for Software Component Search”. In 25<sup>th</sup> International Conference on Software Engineering (ICSE)
- [11] Nutter, D.; Boldyreff, C.; Rank, S.(2003) “An Artefact Repository to Support Distributed Software Engineering”. In: 2nd Workshop on Cooperative Supports for Distributed Software Engineering Processes.
- [12] Ye, Yunwen.(2001) ”Supporting Component-Based Software Development with Active Component Repository Systems”. PhD thesis, University of Colorado.
- [13] Component Source.(2007) <http://www.componentsource.com>.
- [14] Laurillau, Yann; Nigay, Laurence. (2002). “Clover Architecture for Groupware”. In Computer Supported Cooperative Work (CSCW’02), New Orleans, Louisiana.
- [15] Ferraiolo, D. F.; *et al.*. (2001) “Proposed NIST Standard for Role-Based Access Control”. ACM Transactions on Information and Systems Security, Vol. 4, No. 3
- [16] Lau, K., (2001), “Component Certification and System Prediction: Is there a Role for Formality?”. In 4th International Workshop on Component-Based Software Engineering (CBSE’01), Toronto, Canada.
- [17] Schuenck, Michael. (2006) “X-ARM: Um Modelo de Representação de Artefatos de Software”. Dissertação de Mestrado, DIMAp-UFRN, Natal-RN.
- [18] Fielding, Roy Thomas. (2000). “Architectural Styles and the Design of Network-based Software Architectures”. PhD thesis, University of. University of California.
- [19] Gerosa, Marco Aurélio. (2006) “Desenvolvimento de groupware componentizado com base no modelo 3C de colaboração” – Tese (doutorado) – PUC – Rio de Janeiro.
- [20] Grinter, R., (2001).From local to global coordination: lessons from software reuse. In: Proc. of the 2001 International ACM SIGGROUP, Colorado, USA.
- [21] Dangelmaier, W., Hamoudia, H. and Klahold, R. (1999) “CIPD – On Workflow-Based Client Integration” in ACM SIGGROUP Bulletin, Vol 20, pag 20-25.
- [22] Sun Microsystems. (2006) “Java™ Platform, Enterprise Edition 5” <http://java.sun.com/javaee/5/docs/API>
- [23] JBoss Reference Guide(2007) <http://docs.jboss.com/jbportal/v2.6/reference-guide/en/html/>
- [24] Grundy J., Wang X. and Hosking J. “Building multi-device, component-based, thin-client groupware: issues and experiences”. In Australian Computer Science Communications. Australian Computer Society, Inc, Darlinghurst, Australia.(2002)
- [25] Stal, M. (2002).“Web Services : Beyond Component Based Computing“. Communications of the ACM, Vol. 45, Issue 110, pp 71-76.
- [26] Hartman, Bret. Flinn , Donald J. Beznosov , Konstantin. Kawamoto , Shirley. (2003) “Mastering Web Services Security”. Wiley Publishing Inc.
- [27] OASIS Standard Specification. (2006). “Web Services Security”. <http://docs.oasis-open.org/wss/v1.1/>
- [28] MySQL 5.0. (2007). “Reference Manual”. Copyright 1997-2007 MySQL AB.
- [29] Greenberg, S. (2007) “Toolkits and Interface Creativity”, Journal of Multimedia Tools and Applications, Volume 32 , Issue 2, pag 139-159.
- [30] Frakes, William. Kang, Kyo.(2005). “Software Reuse Research: Status and Future”. IEEE Transactions On Software Engineering, Vol.31, N° 7, July.

# Mo Porã – Um sistema gerenciador de repositórios distribuídos e colaborativos no ambiente científico da Amazônia

K. J. Amaral Serique, J. L. Campos dos Santos, F. Santos Costa & J. M. F. Maia

Instituto Nacional de Pesquisas da Amazônia – INPA

Experimento de Grande Escala da Biosfera-Atmosfera na Amazônia – LBA

E-mail: {kleber, lcampos, felipesc, jairmaia}@inpa.gov.br

***Abstract.** This paper presents a system to manage distributed and collaborative repositories, developed to support data activities for long term research projects which are in development in Amazonia. The system describes the main computational resources, required by those experiments and the evolution of the collaborative systems, with their characteristics and applications. For that, it is discussed the technology to support collaborative systems available, based on functional requirements and analyzed systems to guide in the development of Mo Porã tool. The tool was designed take into account an analysis method, three tier architecture and a set of interfaces, developed to promote the necessary interaction amongst data bases.*

***Resumo.** Este trabalho apresenta um sistema de gerenciamento de repositórios distribuído e colaborativo desenvolvido para atender atividades de dados em projetos de pesquisas de longa duração, que estão em desenvolvimento na Amazônia. O sistema descreve os principais recursos computacionais que tais experimentos solicitam e a evolução dos sistemas colaborativos, com suas características e aplicações. Para tanto, é discutido as tecnologias de suporte para sistemas colaborativos disponíveis, baseado nos requisitos funcionais e de sistemas analisados para orientar no desenvolvimento da ferramenta Mo Porã. A ferramenta foi projetada considerando um método de análise, uma arquitetura de três camadas e um conjunto de interface desenvolvida para promover interações necessárias entre bases de dados.*

## 1. Introdução

As instituições científicas da Amazônia, a exemplo do Instituto Nacional de Pesquisas da Amazônia (INPA) e Museu Paraense Emílio Goeldi (MPEG), consideram fundamental o gerenciamento de dados por oferecer suporte e facilitar suas atividades de pesquisa por ser imprescindível aos procedimentos de análises e segurança da informação.

Diante do aumento do volume dos dados, compartilhá-los intensivamente reflete as responsabilidades institucionais assumidas, além de propiciar os meios para refinadas evoluções em processos de síntese multidisciplinar de dados.

A tecnologia da informação proporciona diversas formas de compartilhamento de dados, arquivos e conhecimentos. Entretanto, gerenciar grande volume de dados tratados e conhecimentos gerados de forma colaborativa com o objetivo de facilitar sua disponibilidade e acessibilidade aos usuários com eficiência e segurança, ainda é um grande desafio para desenvolvimentos de sistemas colaborativos voltados a ciência e tecnologia. Tais dificuldades devem-se as diferentes abordagens e necessidades de diversos projetos científicos em andamento. No caso das instituições científicas na região amazônica, esta dificuldade é potencializada pelas grandes distâncias que as separam.

Neste contexto, é evidente a necessidade de ferramentas que aportem tecnologias que objetivem o gerenciamento de repositórios de arquivos de múltiplos formatos, das informações sobre usuários (grupos de pesquisa), da classificação dos conteúdos desses grupos, do controle de acesso em nível de configuração do perfil de usuários em ambiente dinâmicos e que permita acesso através da Internet.

Considerando este cenário e buscando contribuir com uma solução robusta, foi desenvolvido uma ferramenta para o ambiente Web, *Mo Porã*<sup>1</sup>, para gerir os dados, arquivos e usuários dos grupos de pesquisa dos Programas LBA e PPBio (Programa de pesquisa em biodiversidade), além da Rede CT-Petro Amazônia, sediados no INPA.

## **2. Demanda de Recursos Computacionais em Experimentos Científicos Integrados na Amazônia**

As atuais iniciativas de investigação científica da Amazônia são apoiadas por diversos recursos computacionais, muitas vezes diferenciados entre si, que complementam e auxiliam seus resultados pesquisados ou ainda para melhorar o entendimento sobre as hipóteses investigadas.

Segundo Chin e Lansing (2004) a colaboração científica é amplamente focada no compartilhamento e análise conjunta de dados e resultados. Buscando contribuir com o desenvolvimento de ferramentas que encapsulem a amplitude de recursos computacionais dos projetos e programas citados, verifica-se a clara necessidade de compartilhamento de dados e informações e possibilitando a aplicação futura aos demais projetos existentes.

O programa científico LBA estuda as complexas interações da biosfera com a atmosfera na Amazônia. Trata-se, de fato, de um grande programa de pesquisas, liderado pelo Brasil e com cooperação científica internacional, composto por mais de 130 projetos de pesquisa, financiados por várias agências nacionais e internacionais, *(com destaques para a NASA e a Fundação Nacional de Ciência, dos EUA, a Comissão Européia, o IAI – Instituto Interamericano de Pesquisas sobre Mudanças Globais, etc)* [Luizão *et al* 2005].

Este experimento conta com diversos recursos computacionais para atender a necessidade deste grande programa de pesquisa. Os recursos incluem: infra-estrutura para automação de escritório, acesso de qualidade a Internet, servidores de banco de dados e de páginas Web, ferramentas de análise de dados e de modelagem climáticas, executadas em supercomputador vetorial e também em arquitetura de cluster, todos

---

<sup>1</sup> A origem da expressão *Mo Porã* vem do guarani que significa guardar em local seguro.

suportados por uma rede de alta velocidade e de alto desempenho, devido aos grandes volumes de dados utilizados.

Outra iniciativa desafiadora, a Rede CTPetro Amazônia, foi criada como uma Rede Temática Cooperativa entre instituições de Ensino Superior e de Ciência e Tecnologia na Amazônia, cujo objetivo é o de intensificar a troca de informações, conhecimentos, intercâmbio de profissionais, treinamento e capacitação, obtenção e divulgação de novos conhecimentos que permitam identificar, avaliar, eliminar ou minimizar os efeitos negativos ao meio ambiente, das atividades de prospecção e transporte do gás natural e petróleo na Amazônia [Rede CTPetro Amazônia 2006]. Para isso, conta com recursos computacionais que vão de estações de trabalho, acesso a Internet, servidores Web e de dados, arquitetura distribuída cliente servidor e banco de dados objeto-relacional.

Mais recentemente, o Programa de Pesquisa em Biodiversidade – PPBio, iniciado em 2004, tem como principais metas promover o desenvolvimento de pesquisa, a formação e capacitação de recursos humanos, além do fortalecimento institucional na área da pesquisa e desenvolvimento da diversidade biológica, em conformidade com as Diretrizes da Política Nacional de Biodiversidade. Este programa utiliza recursos computacionais voltados ao armazenamento de grande volume de dados, plataforma cliente servidor, servidores de dados e Web e ainda ferramentas para gestão de acervos biológicos.

Para capturar as necessidades comuns do trabalho desenvolvido pelos diversos grupos de pesquisas e dos projetos descritos acima, foi iniciada a primeira fase do desenvolvimento do Sistema Mo Porã, com o objetivo de oferecer aos coordenadores e pesquisadores dos grupos de pesquisa um ambiente colaborativo envolvendo aplicações de repositórios de arquivos compartilhados pelos membros dos grupos, ferramentas de gestão de informação sobre os grupos de pesquisa, dos seus respectivos membros e suas funções no grupo, informações sobre subgrupos e das instituições envolvidas, além dos pesquisadores poderem contar com ferramentas de comunicação instantânea para estabelecer comunicação com os demais membros de seus grupos de pesquisa que possam estar distantes geograficamente.

### **3. Sistemas Colaborativos**

A colaboração é um princípio de trabalho em conjunto que produz confiança, integridade e resultados através do verdadeiro consenso, propriedade e alinhamento de todos os aspectos da organização [Sarmiento 2002]. Colaborar é de fato o ato de trabalhar e compartilhar informações entre pessoas ou grupos de pessoas. O processo de colaboração inicia-se quando há a possibilidade de compartilhamento de idéias, recursos e coordenação dos esforços de trabalho.

Na evolução dos Sistemas colaborativos têm-se destaque a Automatização dos escritórios (*Office Automation*) na década de 70, com a necessidade de procurar formas de suporte ao trabalho administrativo de grupos e organizações.

Outro destaque no meio colaborativo é o CSCW (Trabalho Cooperativo Auxiliado por Computador) surgiu por volta da década de 80 e que segundo Lima (2003), tem objetivo de permitir a eficácia dos grupos existentes, dentro das estruturas naturais da organização. Lima (2003) ainda define CSCW como a atividade em que as

pessoas realizam com a ajuda de computadores interligados através de redes de computadores.

Já o termo Groupware pode ser definido como um conjunto de ferramentas que objetivam o aumento da produtividade do trabalho colaborativo, de acordo com [Gerosa *et all* 2005] são difíceis de construir e de manter, pois possuem as dificuldades técnicas de aplicações distribuídas e de sistemas multi-usuário, além de envolverem aspectos multidisciplinares em seu projeto.

O modelo 3C de colaboração, segundo Fuks *et all* (2003) relata que para colaborarem, os indivíduos têm que trocar informações (se comunicar), organizar-se (se coordenar) e operar em conjunto num espaço compartilhado (cooperar). Este modelo é de grande valia para o entendimento para o desenvolvimento de sistemas colaborativos. O modelo 3C pode ser aplicado na Engenharia de Groupware onde é utilizado durante as fases de análise, separando os requisitos de comunicação, coordenação e cooperação.

Em geral pode-se afirmar que um Groupware é um software que suporta CSCW. Exemplos de aplicações de Groupware é o correio eletrônico, sistemas de mensagens curtas, grupos de discussão, entre outras.

#### 4. Tecnologias de Suporte a Sistemas Colaborativos baseados na Web

Com a evolução da *Word Wide Web* (rede de alcance mundial) diversos sistemas de informação passaram por adaptações para funcionarem com uma interface Web, conhecidos como sistemas baseados na Web (*web application*). Os sistemas baseados na Web foram desenvolvidos para serem acessados por meio de qualquer navegador de paginas Web. Existem também os sistemas desenvolvidos com interfaces gráficas que funcionam instalados no computador do usuário e muitas vezes dependem de algum tipo de sistema operacional para funcionar corretamente. Assim os sistemas baseados na Web têm uma maior portabilidade e disponibilidade do que a de sistemas de interface gráfica.

Os sistemas colaborativos baseados na Web devem lidar com diferentes situações, por exemplo, idiomas, culturas e fusos horários. A Figura 1 apresenta uma infra-estrutura computacional em camadas de serviços. Geralmente essas camadas se referem à rede de computadores, no caso a Internet, banco de dados e sistema operacional.

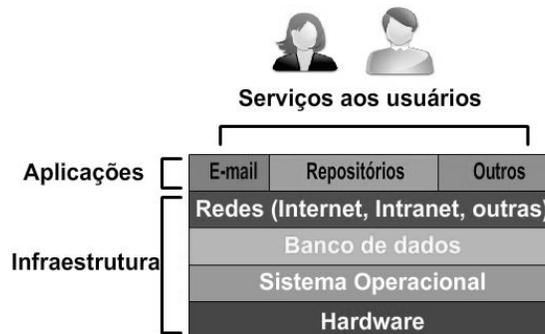


Figure 1. Infra-estrutura de apoio a sistemas colaborativos baseados na Web.

Os sistemas colaborativos centralizados concentram a um único servidor central. Normalmente todas as funcionalidades ficam armazenadas neste servidor, e na interface

com o usuário poderá conter algumas funcionalidades locais de pequenos processos como de manipular algum arquivo. Os sistemas colaborativos descentralizados possuem um conceito diferenciado em relação aos centralizados, onde os usuários possuem bases locais que são sincronizadas com o servidor central, podendo ser usado sem estar conectado ao servidor central. Os sistemas híbridos funcionam de forma centralizada e descentralizada.

Os Sistemas baseados na Web ou *WebApps* são aplicações interativas, portáteis e seguras possibilitando criar ambientes virtuais seguros e robustos, embora utilize a Internet como a principal meio para envios de trocas de dados, podendo ser acessíveis pelos usuários através de aplicativos denominados de navegadores Web que se comunicam com seus respectivos servidores, fazendo requisições de páginas e *downloads* de informações e arquivos, hoje bastante utilizados devido a sua grande portabilidade de acesso imediato.

Os sistemas de interface gráfica são em geral aplicativos que rodam no computador do usuário e assim fornecendo acesso ao sistema através do mesmo. Os sistemas de interface mista podem ser acessados via o software gráfico ou através do navegador Web.

Hoje em dia, *WebApps* evoluíram para ferramentas computacionais sofisticadas que não fornecem somente funções isoladas para o usuário final, mas também são integradas com banco de dados corporativos e aplicações de negócio [Pressman 2006].

## **5. Requisitos para o desenvolvimento**

A engenharia de requisitos estabelece uma base sólida para o projeto e a construção de sistemas automáticos. Sem ela, o software resultante tem uma alta probabilidade de não satisfazer às necessidades dos clientes [Pressman 2006].

Os requisitos de sistema são de uma forma ampla as propriedades em que um sistema de computação deverá realizar ou atribuir valor aquela necessidade. A engenharia de software trata requisitos de sistemas como uma condição ou capacidades necessárias para o usuário solucionar um problema ou alcançar um objetivo.

Os principais requisitos definidos para o desenvolvimento do Mo Porã incluem:

1. Ser desenvolvido utilizando recursos e tecnologias de sistemas livres;
2. O cadastro de novos usuários é aberto à comunidade em geral;
3. O usuário só participa de um grupo se o coordenador ou criador daquele grupo o adicionar ao mesmo;
4. Os usuários de forma geral podem solicitar a criação de novos grupos, porém somente os administradores controlam a criação dos mesmos;
5. Os administradores são podem utilizar todas das funcionalidades do sistema;
6. O sistema deve ser capaz de carregar vários volumes de arquivos de uma só vez;
7. O sistema deve ser capaz de descarregar vários volumes de arquivos de uma só vez;
8. O sistema deverá ser portátil e acessível via Internet;

- 9.O sistema deve possuir uma interface que permita aos administradores visualizarem estatísticas de uso do sistema;
- 10.O sistema deve garantir segurança nos dados transmitidos pela Internet.
- 11.Busca e desenvolvimento de uma solução em banco de dados distribuídos para unir e distribuir as bases Mo Porã em seus sites e, ainda promover a integração de novos sites;
- 12.Integração a solução do Sigepro no componente de gestão de informações sobre os grupos de pesquisa;
- 13.Integração ao Geonetwork para catalogação dos arquivos e criação dos seus metadados;
- 14.Ferramentas de comunicação e conferência para comunicação de usuários dentro do sistema, para o fim de se possuir mais um meio de comunicação entre os usuários Mo Porã.

Os requisitos juntamente com a federação dos sites Mo Porã foi dado início ao aprimoramento do sistema onde foi abordada a integração do sistema com o Sigepro, um sistema de georeferenciamento de projetos desenvolvido no Ministério do Meio Ambiente (MMA), e como os repositórios possuem arquivos de experimentos científicos e outros documentos, está sendo integrado um sistema de gerenciamento de metadados o Geonetwork.

O Sigepro criado em 2004 pelo Ministério do Meio Ambiente, permite que sejam cadastrados Secretarias, Programas e Projetos de forma hierárquica, além de georeferenciar a atuação de um determinado projeto. As informações que podem ser encontradas no Sigepro são relativas às pessoas envolvidas, recursos financeiros, parcerias e convênios, bibliografias, agenda entre outros.

O Geonetwork é um catalogador de metadados geográficos. Permite gerenciar metadados de acordo com os padrões ISO 19115, ISO 19139 e FGDC. Permite a realização de buscas por palavras-chave ou região e pode comunicar-se com outros servidores utilizando busca remota. Pode ser utilizado como uma interface de consulta sobre os metadados dos dados do programa.

## **6. Projeto Mo Porã**

O sistema Mo Porã foi iniciado a partir da necessidade de se compartilhar arquivos e dados entre pesquisadores e colaboradores em grupos de pesquisa dentro do Programa LBA. Como esta necessidade era bastante comum com outros grandes projetos de pesquisa do INPA foi implantando também nos projetos PPBio e na Rede CTPetro Amazônia.

Surgiram novas abordagens dentro do desenvolvimento do Mo Porã, dentre as principais é a de torná-lo uma ferramenta que promovesse um ambiente de colaborativo de compartilhado de informações dos grupos de pesquisa para facilitar a troca de informações entre os pesquisadores destes projetos, pois existem muitos projetos sendo executados e não se tem uma comunicação entre os grupos por não se saber de forma mais acessível onde os grupos de pesquisa estão atuando.

Contudo a ferramenta está sendo desenvolvida com a finalidade de se tornar um sistema capaz de reunir ferramentas colaborativas entre os usuários em um ambiente multidisciplinar e interativo dentro das perspectivas dos projetos científicos. Além de poder ser adicionada novas funcionalidades de necessidades específicas dos grupos de pesquisa.

### **6.1. Arquitetura do sistema**

A arquitetura tecnológica utilizada no projeto Mo Porã, em sua segunda versão, conta com o Servidor Web Apache, a linguagem de programação PHP, o banco de dados Postgresql, o replicador de bases de dados *Slony-I*, *applets* em Java e tecnologias combinadas para interação com usuário Ajax. Na primeira versão não estão presentes o uso do replicador de bases de dados e dos recursos em Ajax.

Na Figura 2 exemplifica as camadas de tecnologias utilizadas para o funcionamento do Mo Porã, onde temos o Navegador Web, Máquina Virtual Java, *applets*, os protocolos de rede http ou https, Servidor Web com PHP e Banco de dados. Os *applets* são pequenos aplicativos escritos em Java que executam na máquina do usuário e necessita de máquinas virtuais Java para funcionar corretamente. O *Javascript* é uma linguagem interpretada pelo navegador Web, onde combinado com a metodologia Ajax que utiliza as tecnologias de XML e *Javascript* que funciona fazendo com funções em *Javascript* que realizam requisições ao PHP que retorna para as funções em *Javascript* informações em XML que serão tratados e inclui dentro da página do usuário, assim eliminando a necessidade de carregamento total da página e carregando somente partes dela quando necessário. O protocolo http é utilizado nas requisições de informação entre o Navegador Web, na máquina do usuário, com o Servidor Web. O PHP é uma linguagem de script que responde as requisições em PHP feitas pelo navegador Web que é capaz de se comunicar com bancos de dados. O PostgreSQL é um banco de dados Objeto relacional muito robusto capaz de ser replicado através do *Slony-I* que é do tipo assíncrona onde toda transação de atualização feita na base *master* se propaga em uma outra transação para as bases *slave* assim todas as bases são replicadas e ficam idênticas a base *master*.

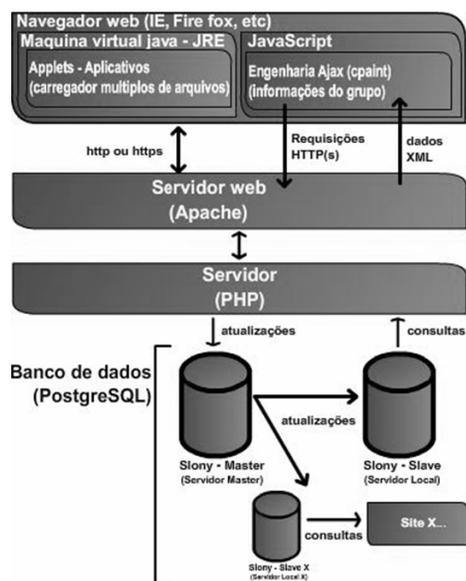


Figura 2. Infra-estrutura de apoio do sistema Mo Porã.

O Mo Porã conta com a possibilidade de ter um cadastro aberto a qualquer pessoa, no entanto, o usuário só acessará aos repositórios de arquivos se for membro de algum grupo de pesquisa ou de trabalho. Cabe somente ao coordenador de cada grupo as atividades de adicionar usuários ao seu grupo, especificando a cada um a sua política de acesso aos arquivos dos repositórios. A criação de novos grupos de pesquisa pode ser solicitada por qualquer usuário, mas somente os administradores do sistema que autorizam a criação dos novos grupos. Os administradores possuem o controle geral do sistema podendo obter os mesmos privilégios dos coordenadores dos grupos, por exemplo, adicionar usuários, remover usuários, dar privilégios de acesso aos usuários.

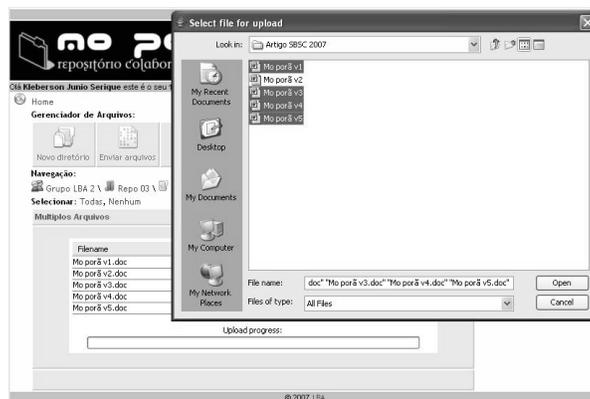
Diferentemente da primeira, os conteúdos de informações dos grupos de pesquisa estão disponíveis a usuários não cadastrados, os usuários visitantes, podem ler e até baixar arquivos públicos dos repositórios.

Com a utilização e o fácil entendimento do sistema, logo surgiram diversas necessidades, por exemplo, dentro do gerenciador de repositórios conta com recurso de carregamento de vários arquivos de uma só vez ao sistema, utilizando um aplicativo, *applet*, escrito em Java, pois os *Browsers* não contam com a capacidade de selecionar vários arquivos. Na Figura 3 é possível observar que somente um arquivo pode ser selecionado assim teríamos que criar mais de uma caixa de seleção para carregar mais de um arquivo e selecionando um de cada vez, para operações de vários arquivos esta solução fica inviável e cansativa por parte do usuário.



**Figura 3. Caixas de seleção por único arquivo.**

Um dos problemas em usar ferramentas HTML é a restrição de uploads de múltiplos arquivos. Entretanto foi encontrado um *applet* que pudesse criar uma lista de arquivos para carregamento em uma única vez de diversos arquivos e com a possibilidade de selecionar mais de um arquivo em uma caixa de seleção (Figura 4). Na primeira versão do sistema utilizamos o *postlet*, um *applet* de licença livre [POSTLET 2007]. Este recurso proporcionou uma maior agilidade para os usuários que tenham necessidades de enviar várias quantidades de arquivos para um diretório de uma única vez, criando uma lista de arquivos e enviar-los de uma única vez.



**Figura 4. Carregamento de múltiplos arquivos.**

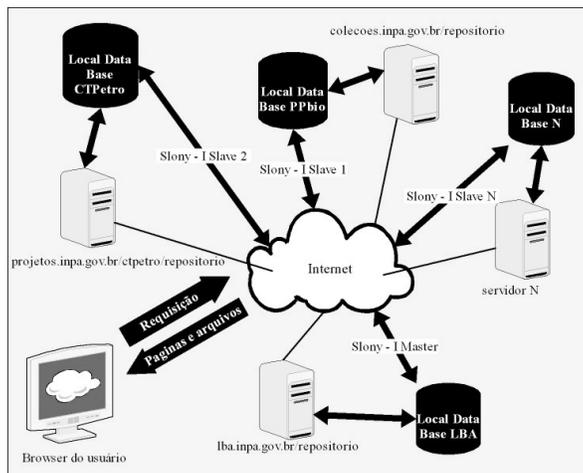
Outro aspecto positivo dentro do recurso de gerenciamento de repositórios é o controle de versões de arquivo, onde cada arquivo pode ser baixado e modificado pelo usuário e carregado, desde que o usuário tenha estes privilégios dentro do repositório, novamente gerando uma nova versão para aquele arquivo, assim não sobrescreveria ou excluiria os registros da versão anterior.

No controle dos grupos de usuários que está sendo desenvolvido que conta com os recursos de georeferenciamento de projetos, recurso herdado da integração com o Sigepro e entre outros recursos.

Um considerável aprimoramento é a união das bases de dados dos três primeiros sites, LBA, Rede CTPetro Amazônia e PPBio, onde é replicada através da *Slony-I*, ferramenta de replicação de banco de dados PostgreSQL, esta solução poderá ser

repassada aos futuros sites Mo Porã, ou seja, além desses três poderão ser inclusos outros sites que queiram participar desta rede de informações e dados compartilhados.

Outra vantagem do sistema é a possibilidade de que um site Mo Porã esteja indisponível, o usuário poderá acessar os seus dados, tanto pessoais como de seu grupo de pesquisa, em um dos outros sites que estejam disponíveis, pois todas as informações serão replicadas. A Figura 5 apresenta a estrutura atual de funcionamento.



**Figura 5.** Arquitetura das Bases de dados dos sites Mo Porã.

A adição de novos sites Mo Porã é um dos atrativos desta arquitetura. Para o desenvolvimento dessa ferramenta foram utilizadas as seguintes tecnologias Web e de banco de dados:

- Banco de dados – o sistema gerenciador de banco de dados é o recurso mais importante deste sistema, pois manipula os aspectos estáticos e dinâmicos da aplicação. Utilizamos o banco de dados relacional Postgresql por se tratar de uma ferramenta robusta e confiável e de licença livre;
- Servidor de páginas Web – foi adotado o Servidor Web Apache nos servidores, seguindo uma orientação de tecnologia já consolidada no INPA. A plataforma é independente do sistema operacional do servidor, além disso, é considerado por especialistas como um dos melhores Servidores Web disponíveis no mercado;
- Linguagem de Programação para Web – A combinação do Apache, Servidor Web, com a linguagem PHP, que é capaz de processar as requisições das páginas permitindo acessar bancos de dados que utiliza recursos de sessões, é muito eficiente, pois permite identificar o usuário que esta acessando a página do sistema. PHP é uma das linguagens mais utilizada na programação para Web. Isso deve-se por ser bastante robusta e de fácil entendimento e uso, além de ser de tecnologia aberta;
- Navegador Web – Com a popularidade da Internet e da navegação de páginas pela Web, surgem vários navegadores de páginas, programas clientes para visualizar os conteúdos de páginas na Web. Entre os produtos os mais adotados pelos usuários da Internet são o Internet Explorer e o Mozilla Firefox. O Mo

Porã foi testado nestes navegadores, podendo ser acessado também pelos demais.

## **7. Considerações finais**

O aporte de diversas tecnologias recentes e abordagens robustas, o Mo Porã tornou-se um sistema colaborativo capaz de realizar trocas de arquivos, publicação de informações dos grupos e meio de comunicação entre os membros dos grupos através das novas ferramentas de comunicação e discussão de idéias. Adicionalmente estarão incluídas as ferramentas de georeferenciamento de informações de projetos, herdada da migração do Sigepro, e também a possibilidade dos pesquisadores publicarem seus metadados, referente aos dados cadastrados no Mo Porã, através da integração com o Geonetwork para gerenciar os metadados dos arquivos cadastrados nos repositórios dos grupos.

Com a conclusão da ferramenta, observou-se uma ferramenta muito eficiente e robusta com novos recursos, além de promover um ambiente flexível para integração de outras ferramentas que venham a ser necessárias para os projetos científicos ou rede de projetos de vasta amplitude.

## **Agradecimentos**

Agradecemos ao INPA e Programa LBA, pelo apoio institucional e financeiro durante o desenvolvimento do sistema. A FAPEAM pelo apoio recebido através do projeto Desenvolvimento e Implementação de uma estratégia para o Gerenciamento de Bio-Metadados baseada na tecnologia XML (eXtensible Markup Language).

## **Referencias**

- Sarmiento, A. M. T, (2002) “Impacto dos Sistemas Colaborativos nas Organizações - Estudo de Casos de Adapção e Utilização de Sistemas Workflow”, Dissertação de Doutorado. Universidade do Minho.
- Lima, W. T. (2003) “Avaliação de usabilidade em sistema colaborativo na área bancária.” São Paulo. Dissertação (Especialização) – Escola Politécnica da Universidade de São Paulo.
- Fuks, H., Raposo, A. B., Gerosa, M.A. (2003). “Do Modelo de Colaboração 3C à Engenharia de Groupware”. IX Simpósio Brasileiro de Sistemas Multimídia e Web - Webmidia (Trilha especial: Trabalho Cooperativo Assistido por Computador), p.445-452. Salvador, BA, Brasil. Novembro.
- Chin Jr G., Lansing C.S. (2004). “Capturing and supporting contexts for scientific data sharing via the biological sciences collaboratory”. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, November 06-10. Chicago, Illinois, USA.
- Gerosa, M.A., Pimentel, M. G., Filippo, D., Barreto, C.G., Raposo, A.B., Fuks, H. & Lucena, C.J.P. (2005) “Componentes Baseados no Modelo 3C para o Desenvolvimento de Ferramentas Colaborativas”, Anais do 5º Workshop de Desenvolvimento Baseado em Componentes - WDBC 2005, 07-09 de Novembro, Juiz de Fora – MG, ISBN 85-88279-47-9, pp. 109-112. Disponível em <http://www.les.inf.puc-rio.br/groupware>

- Luizão, F. B, Nobre, C. A, Manzi, A. O (2005). “Projeto LBA: Estudando as Complexas Interações da Biosfera com a Atmosfera na Amazônia”, Publicado em: Acta Amazônia.
- Gerosa, M. A, (2006) “Desenvolvimento de Groupware Componentizado com Base no Modelo 3C de Colaboração”, Rio de Janeiro.
- Pressman, R. S. (2006) “Engenharia de Software. 6a ed.” São Paulo, McGraw-Hill.
- Rede CTPetro Amazônia (2007) “Apresentação”, <http://projetos.inpa.gov.br/ctpetro>, Fevereiro.
- POSTLET (2007). "Java applet for the transfer of files to an HTTP server". <http://www.postlet.com>. Fevereiro.