UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Christopher Viana Lima

CONTROLADOR DE CONFERÊNCIAS PARA SISTEMAS COLABORATIVOS

Dissertação de Mestrado submetida à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. Roberto Willrich

Florianópolis, novembro de 2007.

CONTROLADOR DE CONFERÊNCIAS PARA SISTEMAS COLABORATIVOS

Christopher Viana Lima

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Dr. Mario Antonio Ribeiro Dantas
Coordenador do Curso

Banca Examinadora

_____

Prof. Dr. Roberto Willrich (Orientador)

_____

Profa. Dra. Roberta Lima Gomes

_____

Prof. Dr. Carlos Barros Montez

_____

Prof. Dr. Frank Augusto Siqueira

Aos meus pais que sempre me apoiaram e ao meu filho Bryan.

AGRADECIMENTO

Aos meus pais Adair e Ivonete e ao Bryan pelo carinho e atenção.

Ao meu orientador, Roberto Willrich, pela dedicação, paciência e confiança depositadas em mim durante este trabalho e em projetos de pesquisa.

A Roberta Lima Gomes e ao Guillermo Hoyos-Rivera pelo apoio e indispensável ajuda para tornar este trabalho possível.

Ao projeto CNPq SIDIE como apoiador do meu trabalho.

Andréa Cesco pela correção de português feita neste trabalho e a sua família em especial a Hiassana Scaravelli pelo grande apoio durante esta jornada.

A todo o pessoal do LaPesD: Denise, Parra, Vinícius e Guilherme. Em especial ao Rafael Speroni e ao Jeferson Rodrigues pela colaboração na implementação do Controlador.

A Elaine pelo carinho e compreensão por esta etapa da minha vida.

À Verinha e ao Hugo da secretaria do curso que sempre são tão prestativos.

Aos meus amigos de bar e música: Marrão, Júnior, Dimy, Andrey, Didi, Adamô, Walmoli. Em especial a Nicole Duarte por ter me auxiliado a terminar o abstract.

Agradeço também a todos que de alguma forma direta ou indiretamente ajudaram neste trabalho.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AGI	Asterisk Gateway Interface
ARA	Asterisk RealTime Architecture
CoLab	Collaborative Browser
CSCW	Computer Supported Cooperative Work
DAC	Distribuição Automática de Chamadas
GPL	General Public License
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
IVR	Interactive Voice Response
LClient	LEICA Client
LEICA	Loosely-coupled Environment for Integrating Collaborative Applications
MGCP	Media Gateway Control Protocol
RTP	Real-time Transport Protocol
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network
SDT	Synchronization Dependency Tree
SIP	Session Initiation Protocol
SMTP	Simple Mail Transport Protocol
UAC	User Agent Client
UAS	User Agent Server
URA	Unidade de Resposta Audível
VoIP	Voice over Internet Protocol
XML	Extensible Markup Language

RESUMO

Trabalhos colaborativos são atividades conduzidas por grupos de indivíduos apresentando diferentes requisitos e tarefas a serem executadas. Em geral, devido à diversidade em termos de tarefas a serem executadas, um único sistema de trabalho colaborativo não consegue atender as necessidades de todos os possíveis tipos de trabalho colaborativo. Conseqüentemente, os usuários precisam usar um conjunto de ferramentas, que satisfaçam suas necessidades. Normalmente estas ferramentas são usadas em paralelo, sem nenhuma integração efetiva. Esta integração permite que diferentes funcionalidades de colaboração sejam dinamicamente combinadas e controladas de acordo com as necessidades. Uma das contribuições neste sentido é o ambiente de integração LEICA (*Loosely-coupled Environment for Integrating Collaborative Applications),* que permite estender as funcionalidades do ambiente colaborativo através da agregação de novas ferramentas.

Um dos elementos fundamentais para um ambiente de colaboração é a comunicação efetiva entre os participantes, implementada por uma ferramenta de conferência. Na perspectiva de compor estes sistemas, uma ferramenta de conferência idealmente deveria permitir a sua integração com outras ferramentas colaborativas. Este trabalho apresenta um controlador de conferências junto a um servidor PBX Asterisk, compondo uma ferramenta que pode ser utilizada em outras aplicações colaborativas, permitindo novas funcionalidades de comunicação. A solução de integração dessas aplicações colaborativas adotadas neste trabalho foi o ambiente LEICA.

Como experimento realizado este controlador foi integrado a um ambiente de co-navegação CoLab para este disponibilizar suporte a conferências.

Palavras-chave: Ambientes Colaborativos, Conferência, VoIP e CSCW.

ABSTRACT

Collaboration activities usually involve a group of people with different needs and task to be executed. Due the diversity in terms of tasks to be performed in different contexts of collaborative work, in general, a single system doesn't fulfill the needs of all possible types of collaborative work. Therefore, users need to use a set of tools that together satisfy their needs. Usually these tools are used in parallel, without any effective integration. This integration allows different functionalities of collaboration to be dynamically combined and controlled according to the needs of each user. One of the contributions in this field is the integration environment LEICA (Loosely-coupled Collaborative Environment for Integrating Applications), which allows extending the collaborative features of the environment through the inclusion of new collaborative tools.

One key element for a collaborative environment is an effective communication among its participants, implemented by a conference tool. In view of these systems, a conference tool should allow its integration with other applications. This work presents a Conference Controller that, integrated to an Asterisk PBX server, consists of a conference tool that can be integrated with other applications, allowing new communication features. The solution for integration of these collaborative applications adopted in this work was the LEICA environment.

For experimentation purpose, this controller has been integrated with the CoLab co-navigation to provide support to audio-conference.

Keywords: Collaborative environment, SIP Conference, VoIP and CSCW.

1 INTRODUÇÃO

A grande demanda de trabalhos em conjunto envolvendo indivíduos geograficamente dispersos fez surgir a necessidade de desenvolver uma variedade de ambientes colaborativos ou sistemas CSCW (*Computer Supported Cooperative Work*). CSCW são sistemas baseados em computador que suportam grupos de pessoas em uma tarefa ou objetivo comum, fornecendo uma interface para ambientes colaborativos [Nielsen, 1996].

Diferentes ferramentas podem ser incluídas em um ambiente colaborativo como sistema de conferência, *chat*, quadro branco e co-navegação. Através de um sistema de conferência os usuários podem se comunicar através de voz e/ou vídeo. No caso do chat, estes usuários podem se comunicar por mensagem instantânea. Já o quadro branco pode ser utilizado principalmente para explicações com base em imagens de fundo (ex. para comentar e realizar inserção de figuras e textos sobre um logotipo de uma empresa que se está criando de maneira colaborativa). As ferramentas de co-navegação (ex. CoLab) permitem que várias pessoas possam navegar em conjunto de forma sincronizada. Muitas áreas podem se beneficiar do paradigma de co-navegação, como: ambientes de aprendizagem à distância, busca colaborativa de informações, navegação em materiais de suporte em vídeo-conferência [Hoyos-Rivera, 2006]:.

Os usuários de um ambiente colaborativo normalmente são um conjunto de pessoas com semelhantes tarefas e necessidades. Uma das maiores dificuldades na criação deste tipo de ambiente é antecipar todas as necessidades de seus potenciais usuários nas mais diversas situações de trabalho colaborativo. Por exemplo, no ensino à distância, todos os participantes estão em uma conferência única, e na maior parte do tempo o professor interage com o sistema e os alunos assistem passivamente. Em alguns momentos o professor pode passar o controle de algumas ferramentas colaborativas para determinados alunos. Já em projetos colaborativos, diversas pessoas podem efetivamente colaborar em um projeto, cada um solicitando o controle de determinadas ferramentas. Neste último, existe a possibilidade de se criar grupos de trabalhos, os quais temporariamente podem formar pequenas sessões de conferência para realizar trabalhos individuais visando ao final

uma meta conjunta. Cada um destes cenários pode requerer diferentes ferramentas de colaboração, inclusive com diferentes formas de operação.

É muito difícil um ambiente colaborativo único atender todas as necessidades dos mais diversos cenários de trabalho colaborativo. Conseqüentemente, para atender todas as suas necessidades, os usuários utilizam de forma paralela um conjunto de ferramentas colaborativas. Estas ferramentas geralmente não são integradas de maneira automática, sem tirar vantagem uma das outras. Neste sentido, é de responsabilidade dos usuários manualmente realizar todas as operações necessárias quando um evento em uma ferramenta requerer uma ação em outra ferramenta. Por exemplo, se um grupo de participantes formar um grupo de trabalho para utilizar um editor de texto compartilhado, é de responsabilidade do usuário criar uma áudio-conferência formada por este grupo.

A integração entre as aplicações colaborativas pode trazer significantes benefícios para os usuários, como permitir que estes não precisem interagir com todas as ferramentas e sim se preocupar unicamente com a sua participação no ambiente colaborativo.

A fim de conseguir tratar os problemas abordados, [Gomes, 2005] propôs LEICA (*Loosely-coupled Environment for Integrating Collaborative Applications*), um ambiente de integração de aplicações colaborativas fracamente acopladas. Apoiando-se na tecnologia de serviços Web e em um sistema de notificação de eventos, diferentes aplicações colaborativas podem interagir através da troca de informação. Uma vantagem importante de LEICA é que, uma vez integrada ao ambiente, essas aplicações mantêm suas autonomias de execução. Além disso, aplicações podem ser integradas e retiradas do ambiente sem comprometer a execução deste último, garantindo assim uma maior flexibilidade. A abordagem fracamente acoplada proposta pelo LEICA permite tratar dois problemas normalmente encontrados em ambientes colaborativos: não requer uma integração verdadeiramente semântica das aplicações colaborativas; e uma vez integradas ao ambiente, essas aplicações mantêm sua autonomia.

Uma das ferramentas essenciais para o trabalho colaborativo é a conferência, que possibilita a comunicação com áudio e/ou visual entre os membros do trabalho colaborativo. Existem diversas ferramentas de áudio-conferência que podem ser usadas em ambientes colaborativos. Um exemplo é o servidor PBX Asterisk [Asterisk, 2007]. Um

servidor PBX convencional fornece serviços como caixa postal de voz, conferências, resposta de voz interativa e filas de chamadas. O Asterisk como um PBX também possui os mesmos tipos de serviço, com a vantagem de gerenciar, além das ligações convencionais, ligações de voz sobre IP (VoIP) como também suporte a outras aplicações externas para adicionar novas funcionalidades ao servidor.

Durante diversos cenários de trabalho colaborativo é normal que os membros mudem de conferências no decorrer da sessão colaborativa. Se não há uma integração entre as aplicações, os usuários precisam manualmente entrar e trocar de conferências, dificultando, portanto, o trabalho colaborativo. Com uma integração da ferramenta de conferência às demais aplicações colaborativas, o gerenciamento das conferências poderia ser feito de forma dinâmica. Esta integração traz diversos benefícios, no sentido que os participantes se preocupem mais com as atividades do trabalho colaborativo.

1.1 Objetivos

1.1.1 Objetivo Geral

O presente trabalho propõe um Controlador capaz de gerenciar e controlar conferências, visando ser uma das ferramentas colaborativas a ser integrada na criação de ambientes colaborativos personalizados, que atendam as necessidades dos usuários em trabalhos colaborativos específicos. Neste trabalho, foi adotado o ambiente LEICA para a integração dessas aplicações. O Controlador consiste de uma aplicação Java que controla as operações de um PBX IP Asterisk. Este controlador oferece uma API direcionada à integração desta aplicação ao ambiente LEICA. Também foi desenvolvido um *Wrapper* LEICA para o controlador.

Para a validação da proposta, foi implementado um sistema de co-navegação via integração da ferramenta de co-navegação CoLab [Hoyos-Rivera, 2005] e do nosso controlador de conferências. Desta forma, operações realizadas no CoLab (ex., operações de sincronização) podem conduzir à realização automática de operações no serviço de conferência do controlador e vice-versa.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho foram os seguintes:

- Implantar e testar um servidor PBX IP para suporte a conferências;
- Pesquisar e integrar uma ferramenta para conferência ao servidor PBX IP;
- Definir os requisitos para um controlador de conferências adaptado a ambientes colaborativos.
- Implementar e validar o gerenciador de conferência composto de uma API para integração com outras ferramentas colaborativas.

1.2 Organização do Trabalho

O presente trabalho está dividido em 6 capítulos.

- Capítulo 1: apresenta uma introdução ao trabalho desenvolvido, bem como os objetivos do mesmo;
- Capítulo 2: apresenta alguns conceitos associados a trabalho colaborativo. Além disso, ele apresenta algumas ferramentas colaborativas usuais, e em especial a ferramenta de co-navegação CoLab.
- Capítulo 3: introduz conceitos relativos a VoIP e apresenta o PBX IP Asterisk;
- Capítulo 4: apresenta a solução de integração de ferramentas colaborativas LEICA.
- Capítulo 5: apresenta o controlador de conferências para ambientes de trabalho colaborativo;
- Capítulo 5: trata de um estudo de caso com o ambiente de co-navegação integrado ao controlador de conferência;
- Capítulo 6: apresenta as conclusões relativas ao trabalho, bem como algumas sugestões para trabalhos futuros.

2 TRABALHO COLABORATIVO

CSCW pode ser definida como sistemas baseados em computador que suportam grupos de pessoas engajadas em uma tarefa ou objetivo comum e que fornecem uma interface para um ambiente compartilhado [Nielsen, 1996]. O CSCW é a área que estuda a concepção e a utilização de aplicações colaborativas também chamadas de *groupware*. *Groupware* são aplicações colaborativas que permitem a vários usuários trabalharem em conjunto através de uma infra-estrutura tecnológica, em uma intranet ou Internet. Existem diferentes tipos de *groupware* como: ferramentas de comunicação, troca e compartilhamento de dados, videoconferências, quadro branco compartilhado etc.

Este capítulo tem como objetivo apresentar alguns conceitos relacionados com trabalho colaborativo e introduz algumas aplicações colaborativas. Em particular, apresenta a ferramenta de co-navegação colaborativa CoLab [Hoyos-Rivera, 2005].

2.1 Conceito de Sessão Colaborativa

O conceito de sessão é utilizado geralmente em CSCW para estruturar e organizar o trabalho colaborativo de um grupo de usuários [Gomes, 2006]. Segundo [Haake et al, 1999], uma sessão é definida por:

- Um grupo de usuários (ou agentes de software representando eles);
- Um espaço de trabalho comum nos quais estes usuários trabalhem;
- Um modo de cooperação específico usado por estes usuários. [Haake et al, 1999] identifica três modos de cooperação: trabalho individual, trabalho assíncrono, trabalho heterogêneo fracamente acoplado, trabalho heterogêneo fortemente acoplado, e trabalho homogêneo fortemente acoplado.

Desta forma, as aplicações colaborativas devem permitir criar e destruir uma sessão, juntar e deixar uma sessão, e selecionar e alterar o modo de colaboração de uma sessão.

2.2 Classificação das Aplicações Colaborativas

Existem diversas classificações para aplicações colaborativas disponíveis na literatura. Foi escolhido apresentar duas classificações que são consideradas essências na literatura do domínio: uma se refere a características temporais e espaciais, e a outra à categoria funcional da aplicação.

2.2.1 Classificação Espaço x Tempo

O espaço e o tempo constituem as dimensões mais correntes nas classificações de aplicações de colaboração [Ellis et al, 1991]:

- O espaço se refere à distância geográfica que separa os usuários da aplicação. Por exemplo, os membros de uma reunião podem estar em um mesmo recinto ou estar situados em um lugar diferente.
- A dimensão de tempo caracteriza o tipo de interação entre os usuários. Os membros de um grupo podem interagir ao mesmo tempo, onde as ações dos participantes são imediatamente transmitidas aos outros. Este tipo de interação é conhecido como síncrona.

Existe também a possibilidade de um grupo de usuários interagir em momentos diferentes, onde as ações ocorrem em um espaço de tempo. Este tipo de interação é conhecida como assíncrona. Neste caso é importante que as atividades do grupo sejam guardadas permanentemente, para que seus membros possam visualizar o histórico de ações que foram efetuadas antes de sua chegada.

A tabela 1 mostra a matriz espaço X tempo [Johansen, 1988]. Segundo [Grudin, 1994] existem numerosas críticas no que diz respeito a esta classificação e várias propostas foram desenvolvidas para refinar esta classificação como em [Nunamaker et al, 1991] .

	Síncrono	Assíncrono
Mesmo Lugar	Interação face a face	Interação assíncrona
Lugares Diferentes	Interação distribuída síncrona	Interação distribuída assíncrona

Tabela 1 - Matriz espaço x tempo. Fonte: [Ellis, 1991]

2.2.2 Categorias Funcionais das Aplicações Colaborativas

Muitos autores escolhem classificação de aplicações colaborativas por domínios de aplicação, onde uma lista de categorias funcionais é definida para agrupar as aplicações colaborativas [Bernard, 2004] [Laurillau, 2002] [Terzis, 1999]. [Kyng et al, 1991] propõe a classificação dos sistemas colaborativos em quatro classes:

- **Sistemas de Mensagens**: Assíncronos (ex. e-mail).
- **Sistemas de Conferência**: Difere do anterior pois padroniza a forma como as mensagens são organizadas (ex. vídeo-conferências e *newgroups*).
- **Sistemas de Coordenação**: Dirigem o problema de integração e ajuste dos esforços de trabalhos dos indivíduos através da realização de um objetivo comum (ex. controle de versão, calendários eletrônico). Também é assíncrono.
- **Sistemas de Autoria Colaborativa e Argumentação**: Auxiliam a cooperação necessária entre autores na produção de documentos (ex. Wikis). Funciona de forma Assíncrona

2.3 Aplicações Colaborativas

Existem diversas aplicações colaborativas. A seguir são apresentados alguns exemplos de aplicações para compor ambientes de trabalho colaborativo síncronos:

- **Aplicações de mensagem instantânea**: São muito usadas atualmente por determinados grupos de usuários como, por exemplo, servidores dedicados a

jogos. Elas também são muito usadas para comunicação interna em empresas e instituições. Alguns exemplos deste tipo de aplicação incluem ICQ, Jabber e MSN.

- **Aplicações de chat**: Possibilitam o debate em tempo real, via mensagens de texto, de algum tema de interesse dos participantes. Como exemplo de ferramentas se pode citar o Babylon [Babylon, 2007], IRC.
- **Aplicações de conferência**: Aplicativos de conferência podem utilizar tanto vídeo como áudio para debates em tempo real de assuntos em comum. Algumas aplicações deste tipo incluem o NetMeeting e CU-SeeMe.
- **Aplicações de co-navegação**: As aplicações de co-navegação permitem a um grupo de usuários navegar pela Web em conjunto, cada um em seu próprio computador. Como exemplos, temos o co-navegador CoLab [Hoyos-Rivera, 2005] e CoBrowser [Maly et al, 2001].

As duas próximas seções detalham duas aplicações colaborativas importantes no contexto deste trabalho, as aplicações de conferência e de co-navegação.

2.4 Aplicações de Conferência

A função básica de um serviço de conferência é permitir ao usuário manter uma conversa com mais de uma pessoa ao mesmo tempo. A importância deste tipo de aplicação é proporcionar a comunicação entre os participantes sem que tenham que utilizar chat e mensagem instantânea. A vantagem em relação a estas aplicações citadas é que o usuário não precisa interagir com o teclado ou mouse durante a sua utilização, deixando mais tempo livre para utilizar outras aplicações. Outra vantagem das aplicações de conferência é permitir uma emulação da conversação face-a-face, possibilitando a transmissão da entonação de voz, pausas e outros.

Como requisitos de um sistema de conferências, este deve permitir:

- **Gerenciar conferências**: Criar e modificar conferências e efetuar o encerramento das mesmas.

- **Gerenciar usuários**: Possibilitar a adição e expulsão de participantes das conferências.
- **Controle de palavra** (*floor control*): Permitir o controle de palavra na conferência, como quem tem o papel de locutor e ouvinte ou apenas ouvinte. Isto será visto na seção 2.4.2.

Alguns dos sistemas de conferência disponíveis para trabalhos colaborativos são: [WebExMeetMeNow, 2007], [GoToMeeting, 2007] e [HearMe, 2007]. Geralmente estas aplicações fornecem ferramentas de áudio e/ou videoconferência proprietárias permitindo a comunicação junto a outras aplicações (ex. apresentações em slide, navegador convencional), mas sem integração a ambientes colaborativos.

2.4.1 Tipos de Conferência

As conferências podem se comunicar de três formas:

- **Ponto-a-ponto** (ou *unicast*): É a configuração que permite apenas a comunicação entre dois dispositivos, cada um na extremidade do enlace.
- **Multiponto**: A conferência multiponto funciona com várias conexões ponto-a-ponto, o fluxo de cada cliente é combinado e redistribuído de volta para todos os participantes.
- **Multicast**: No multicast o fluxo de dados pode ser entregue a vários destinatários ao mesmo tempo.

2.4.2 Controle de Palavra

O controle de acesso (*floor control*), ou controle de palavra, permite a um participante se dirigir a outro quando estiver de posse de palavra [Dommel, 1997]. Existem diferentes políticas de controle de acesso que poderiam ser aplicadas a conferências. De maneira geral, existem dois tipos básicos de controle de palavra:

- **Sem controle**: É o caso das conferências abertas, onde todos os participantes podem ouvir e falar a qualquer momento. Este é mais indicado para reuniões

de um pequeno grupo de indivíduos, e a organização da reunião é dependente do nível de respeito social/profissional dos indivíduos.

- **Controlada por um Moderador**: Neste caso, um dos membros da conferência é escolhido como moderador. Este moderador pode gerenciar a posse da palavra, podendo passar esta posse a um ou mais membros da conferência. Ele também pode retirar a posse da palavra de algum dos membros.
- **Controle baseado no perfil do usuário**: Neste caso, é associado um perfil a cada usuário. Este perfil contém os direitos do usuário em termos da posse de palavra, podendo o usuário ser o moderador, ouvinte, locutor-ouvinte ou locutor-apenas. O moderador possui o direito de escutar e falar, cabendo a ele a responsabilidade de excluir pessoas da conferência, limitar o número de participantes, modificar o perfil dos usuários e incluir senha para o ingresso na mesma. O ouvinte apenas escuta a discussão na conferência sem poder integrar com os outros. O locutor-ouvinte pode falar e ouvir.

2.4.3 Modos de Ingresso a Conferências

[Rosenberg, 2000] define vários modelos de conferências, dentre os quais os mais relevantes são a conferência *dial-in* e *dial-out*. Na conferência *dial-in*, os usuários chamam o número da conferência e o servidor de conferência mantém um relacionamento ponto-a-ponto com cada usuário participando da conferência. Neste caso, o servidor multiplexa os áudios dos participantes e envia o áudio a estes. A conferência *dial-out* é uma variação da anterior, onde em vez dos usuários se juntarem manualmente à conferência, o servidor escolhe os usuários que farão parte e realiza uma chamada para cada participante se juntar à conferência.

2.5 Aplicações de Co-navegação Colaborativa

A navegação Web colaborativa vem sendo investigada no contexto de diferentes projetos. Alguns trabalhos, como CoLab [Hoyos-Rivera, 2005] e CBW [Esenther 2002],

propõem ferramentas específicas de co-navegação, sem suporte à comunicação. Outras ferramentas, incluindo CoBrowser [Maly et al, 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 et al, 2003] [Singh, 2004]. Da mesma forma, alguns sistemas de VoIP (Voz sobre IP) [Koskelainen, 2004] [iVisit, 2007] e Call Centers via Web [Yeo et al, 2001] [Kim et al, 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].

A co-navegação colaborativa tem crescido de forma relevante nos últimos anos devido ao potencial de criar novas formas de cooperação entre os usuários Web [Hoyos-Rivera, 2006]. Existem vários modelos e propostas sendo desenvolvidas nesta área. Alguns dos navegadores mais representativos são: CoBrowser [Sidler et al, 1998], E-CoBrowse [Chong et al, 2000] e Let´s Browse [Lieberman, 1999].

Todas as propostas oferecem alternativas para navegação Web na forma de sincronização da navegação. Algumas propostas são baseadas em mecanismo de indexação a qual permite a criação lógica de um grupo de usuários com interesses em comum, facilitando assim a comunicação. Existem também ferramentas colaborativas comerciais que fornecem um conjunto de ferramentas de comunicação para permitir a colaboração.

2.5.1 Colab

CoLab (*Collaborative Web Browsing*) [Hoyos-Rivera, 2006] é um sistema de co-navegação baseado em um modelo simples e poderoso de sincronização de navegação, permitindo uma organização dinâmica de uma sessão de co-navegação em grupos de trabalho. Ele permite a criação 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.

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 1 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 1) 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.

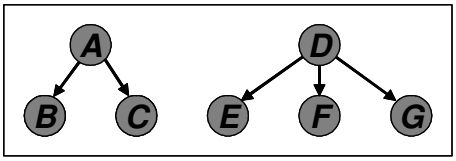

Figura 1 - Cenários de configuração SDTs (Fonte: Hoyos-Rivera, 2006)

A figura 2 apresenta a arquitetura CoLab que segue uma abordagem baseada na utilização de um servidor Proxy. Quando um usuário assíncrono seleciona um hiperlink, esta requisição é enviada ao servidor Proxy CoLab. O servidor satisfaz a requisição e informa a ação aos navegadores membros síncronos do grupo, onde um agente repete as mesmas ações.

O sistema CoLab pode ser dividido em dois módulos: Servidor proxy CoLab e CoLab cliente.

O servidor proxy 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, ou integrar CoLab a outras ferramentas de colaboração.
- **Proxy**: Intercepta as ações de navegação lançadas pelos usuários e pede ao Gerenciador de Sessão que verifique se 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 de 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 de 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.

  A figura 2 ilustra a arquitetura contendo o servidor proxy CoLab e o cliente CoLab.

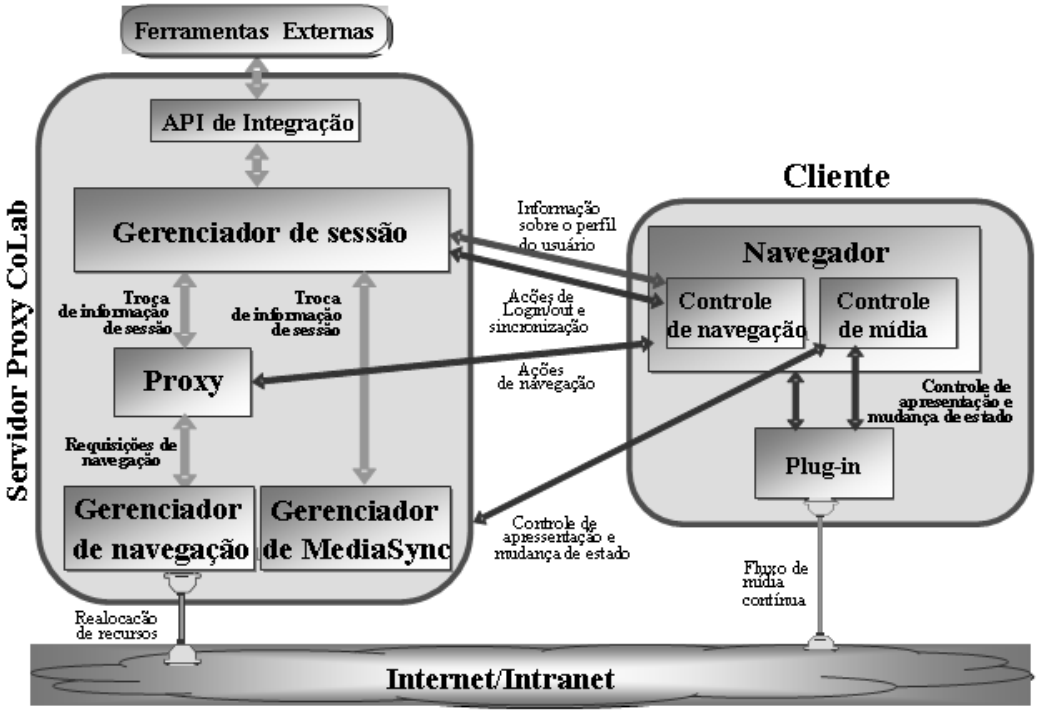

Figura 2 - Arquitetura do CoLab (Fonte: Hoyos-Rivera, 2006)

2.6 Considerações Finais

Neste capítulo foram vistas algumas definições de trabalho colaborativo como também algumas aplicações disponíveis para este ambiente, dando mais destaque à ferramenta de co-navegação CoLab.

CoLab é uma ferramenta de co-navegação que fornece um modelo de sincronização simples e muito flexível que permite a aos membros da sessão criarem dinamicamente e realizar relações de sincronização com outros usuários da mesma sessão. Além disso, CoLab oferece mecanismos permitindo a sincronização de áudio/vídeo durante uma sessão de co-navegação [Hoyos, 2005]. Esta ferramenta não apresenta mecanismos de intercomunicação entre os membros das sessões, o que impede que os mesmos venham a discutir acerca do conteúdo co-navegado.

Algumas ferramentas de conferência também foram vistas como WebExMeetMeNow e HearMe.

3 AMBIENTE DE INTEGRAÇÃO LEICA

Existem duas opções para a implantação de um sistema de trabalho colaborativo. A primeira é utilizar ambientes de colaboração que integram um conjunto de ferramentas colaborativas, como ferramentas de áudio/videoconferência, quadros brancos compartilhados e ferramentas de compartilhamento de aplicações e outros. Embora esta lista de serviços seja grande, em alguns cenários de trabalho colaborativo, estes ambientes não oferecem todos os recursos necessários. Nestes casos é preciso complementar o ambiente de colaboração com ferramentas colaborativas isoladas.

A segunda opção para se criar um sistema de trabalho colaborativo é usar em paralelo diversas ferramentas de colaboração. O conjunto de ferramentas necessárias vai depender dos requisitos dos seus usuários. Neste cenário, as ferramentas oferecem seus serviços de maneira isolada, não havendo nenhuma integração entre elas. Idealmente, este tipo de sistema deveria permitir que diferentes funcionalidades de aplicações existentes fossem dinamicamente combinadas e controladas. A integração de aplicações colaborativas pode trazer significantes benefícios para os usuários, como automaticamente estar no chat da sessão em vigor ou ainda estar na conferência da sessão colaborativa sem ter que trocar manualmente as conferências.

A fim de conseguir tratar esses problemas abordados, [Gomes, 2005] propôs LEICA (*Loosely-coupled Environment for Integrating Collaborative Applications*), um ambiente de integração de aplicações colaborativas fracamente acopladas. Esta abordagem permite tratar dois problemas normalmente encontrados em ambientes de integração: não requer uma integração verdadeiramente semântica das aplicações colaborativas; e uma vez integradas ao ambiente, as aplicações colaborativas mantêm sua autonomia. O grau de interação entre as aplicações depende exclusivamente da API disponível 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.

O objetivo deste capítulo é demonstrar a arquitetura e funcionamento de LEICA e sua integração com outras ferramentas colaborativas.

3.1 Arquitetura de LEICA

A figura 3 ilustra o princípio arquitetural de LEICA. Para integrar as aplicações a LEICA, um *Wrapper* deve ser anexado aos servidores das aplicações colaborativas. Cada *Wrapper* compreende uma interface de serviços Web, permitindo que aplicações colaborativas registrem-se a LEICA como uma aplicação integrada. Assim, na sua utilização o *Wrapper* publica seus serviços em um registro privado UDDI. No caso de aplicações multi-servidor, um servidor mestre deve ser responsável por registar a aplicação. E em aplicações P2P (peer-to-peer) o registro é feito através de um servidor P2P proxy.

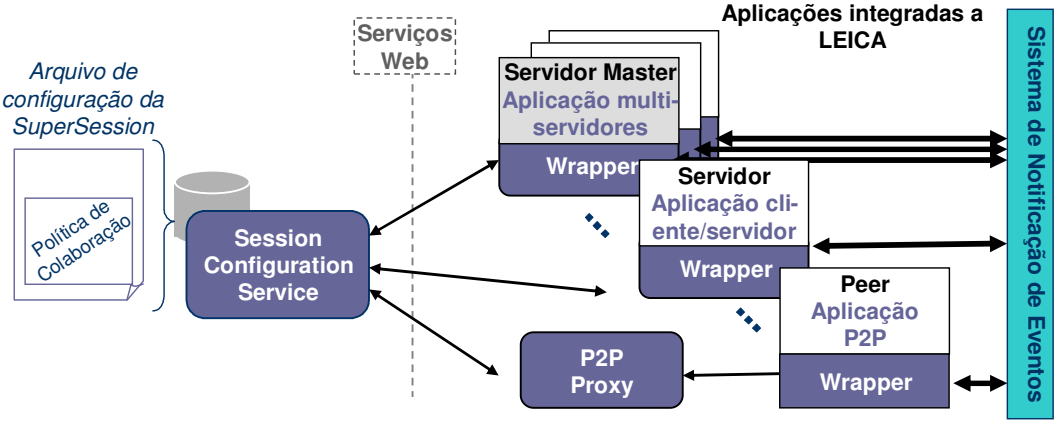

Figura 3 - Princípio arquitetural de LEICA (Fonte:Gomes, 2006)

O *SCS* (*Session Configuration Service*) é um serviço Web usado para configurar e iniciar novas supersessões. Uma Supersessão é uma sessão colaborativa integrada, registrando toda as atividades colaborativas. Dentro do contexto de uma Supersessão, podem existir diferentes Sessões Específicas. Cada Sessão Específica é na verdade uma sessão colaborativa definida por uma aplicação colaborativa (ex., sessão de videoconferência, sessão de co-navegação etc).

Durante a configuração de uma Supersessão, o *SCS* dinamicamente contata 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 da Supersessão. A Política de Colaboração define como as aplicações integradas são coordenadas durante uma Supersessão.

Durante a sua execução as aplicações integradas trocam notificação de eventos ponto-

a-ponto, conforme a atividade colaborativa vai ocorrendo. Os *Wrappers* ficam responsáveis por receber as notificações de eventos e verificar, com base na Política de Colaboração, se este gera alguma ação. Assim o *Wrapper* tem conhecimento se deve requisitar uma ação em uma aplicação integrada. Um fator importante é que LEICA não suporta eventos de baixo nível (ex. uso do mouse, barra de rolagem) ou eventos de alta freqüência de sincronização (ex. posição atual de um objeto). O objetivo é dar apoio a eventos que tenham alguma semântica durante o trabalho colaborativo.

A figura 4 mostra o *framework* de integração de LEICA. A primeira linha mostra os passos da integração do *Wrapper* a aplicação e também seu registro a LEICA. A segunda linha mostra as etapas de configuração para criar a Supersessão. A terceira linha mostra a execução da aplicação junto à Supersessão e seu funcionamento.

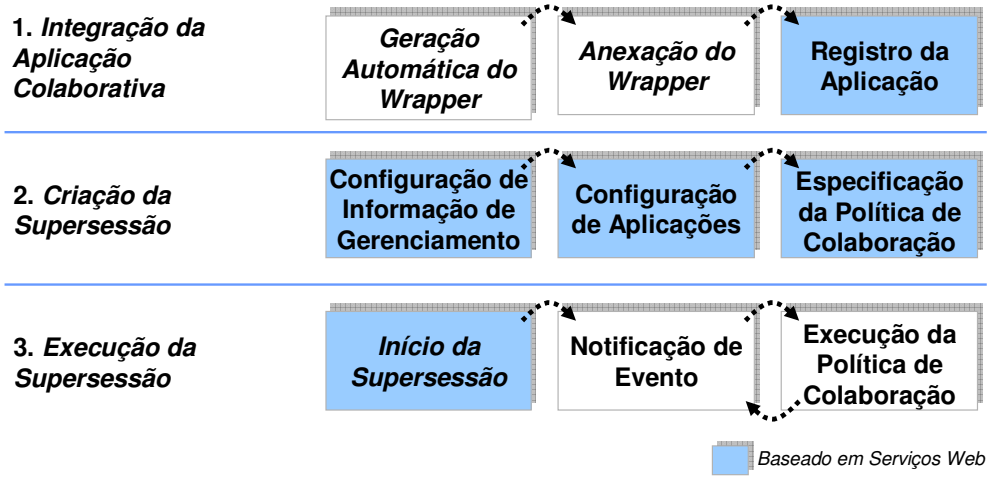

Figura 4 - *Framework* de Integração LEICA (Fonte: Lima-Gomes, 2006)

3.2 *Wrappers* LEICA

LEICA define um módulo específico chamado *API Factory*, que gera um *Wrapper* adaptado para cada aplicação colaborativa. Para tal, este módulo utiliza duas especificações XML: (i) Arquivo de Dados Específicos, que define quais são os dados necessários para criação de sessões convencionais para esta aplicação; (ii) Arquivo de Atributos e API, que define os atributos que caracterizam a aplicação colaborativa e sua API (em termos de tipos de eventos e ações).

O *Wrapper* gerado pelo API Factory deve ainda ser adaptado à aplicação colaborativa. A adaptação de um *Wrapper* consiste em criar o sub-módulo *Application Interface*, o qual representa a interface de comunicação entre ele e a aplicação. Este sub-módulo define uma classe abstrata que deve ser estendida a fim de integrar o *Wrapper* à aplicação.

3.3 Criação de uma Supersessão

Para se criar uma Supersessão é necessário seguir dois procedimentos: configurar o gerenciamento de sessão e configurar as políticas de colaboração. Para configurar o gerenciamento é preciso que dois grupos de informações sejam especificados:

- Informação de Gerenciamento Geral de Sessão (*GSMinfo*): Ele guarda informações de gestão como membros, agendamento e funções gerais do usuário.
- Informação das Aplicações Integradas (*IAinfo*): Define a lista de aplicações integradas a ser usadas pela Supersessão. Para cada aplicação integrada existe uma lista de sessões específicas, onde dados específicos exigidos por esta aplicação para criar sessões é fornecida (ex., uma aplicação de videoconferência fazendo a solicitação de endereço IP multicast).

Uma vez feita esta especificação, deve ser definida a política de colaboração.

3.4 Políticas de Colaboração

A política de colaboração é responsável por definir a semântica de interação das aplicações no ambiente colaborativo. Ou seja, ela define como integrar as diferentes aplicações colaborativas.

As políticas de colaboração são compostas por um conjunto de regras. Cada regra associa um número de eventos a certas ações que devem ser tomadas. Para especificar cada regra é necessário saber quais tipos de eventos as aplicações integradas podem notificar, como também quais ações estas aplicações podem executar.

A fim de prover um modelo mais elaborado para as políticas de regras, [Gomes, 2006] definiu um editor gráfico que permite especificar graficamente as políticas de colaboração. A figura 5 apresenta o conjunto de componentes gráficos (widgets) que podem ser conectados para especificar uma regra de colaboração.

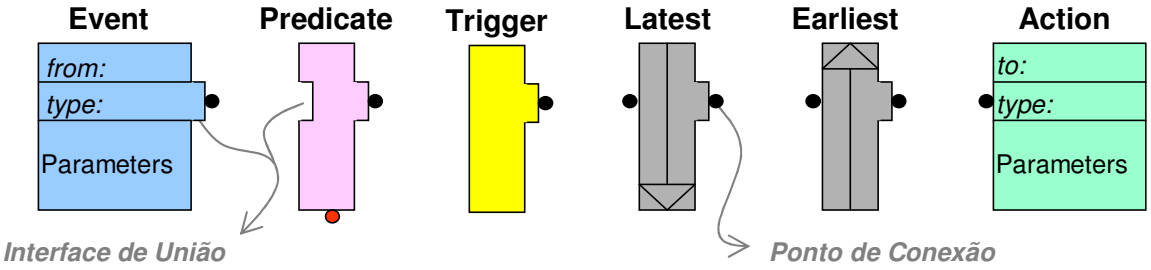

Figura 5 - Componentes Gráficos para as políticas de colaboração (Fonte: Lima-Gomes, 2006)

Com estes componentes gráficos é possível especificar regras simples associando uma notificação de evento à uma ação ou a regras mais complexas combinando diferentes eventos, condições e ações. Os componentes gráficos definidos foram:

- *Event*: Representa uma notificação de evento. Ele possui o campo *Type* para especificar o tipo de evento e o campo *From* que é associada à aplicação colaborativa de origem. Este componente funciona em conjunto com componentes *Actions*, onde um evento gera uma ação respectiva. No campo *Parameters* é possível definir filtros para os valores de parâmetros.
- *Action*: Assim como o componente *Event*, *Action* tem os campos *Type* e *Parameters*. O *Action* recebe do *Event* a solicitação de uma ação conforme a condição especificada no *Event* seja verdadeira. O campo *To* especifica para qual aplicação colaborativa vai a ação referente.
- *Predicate*: Permite especificar uma condição que deve ser satisfeita para ativar uma regra política. Este componente pode ser conectado a qualquer outro componente, exceto ao *Action*. Quando um *Predicate* é associado a um *Event,* ele pode fazer referência aos parâmetros daquele evento. Um *Predicate* é ativado sempre que um componente acoplado for ativado e sua condição for verdadeira.

- *Trigger*: É outro componente que permite definir condições para ativar uma regra política. Diferente do *Predicate,* este não precisa estar conectado a outro componente. Este componente deve ser constantemente monitorado até que sua condição se torne verdadeira e seu controle ativado.
- *Earliest*: Este componente permite a composição de diferentes *Events* e *Triggers* em uma regra política. Neste caso, a regra é ativada quando um dos componentes especificados é habilitado (ex., quando um *Event* é notificado ou quando uma das condições impostas por um *Trigger* se torna verdadeira).
- *Latest*: Tem uma função semelhante ao *Earliest*. Quando os controles são agrupados através do *Latest*, a regra política é ativada após todos os controles serem ativados.

Os componentes gráficos podem ser conectados através dos pontos de conexão ou agrupados pela interface de união. As regras para o uso são [Gomes, 2006]:

- As regras são lidas da esquerda para a direita;
- Apenas componentes sem ponto de conexão ou interface de união à sua esquerda podem aparecer no extremo esquerdo de uma regra;
- Apenas componentes sem ponto de conexão à direita podem aparecer na extrema direita de uma regra de política;
- Toda a regra precisa ser composta de pelo menos um componente *Event* ou *Trigger* e um componente *Action.*

A figura 6 mostra uma regra de integração de duas aplicações (CA1 e CA2) e que associa uma ação a um evento. No exemplo da figura 6, se a aplicação “CA1” notificar um evento do tipo “T1” com o parâmetro b maior que 10, uma ação do tipo “T2” deve ser enviada à aplicação “CA2”. O caractere '%' é um operador de referência, indicando que o parâmetro de ação "y" tem o seu valor copiado do parâmetro "a" do evento T1.

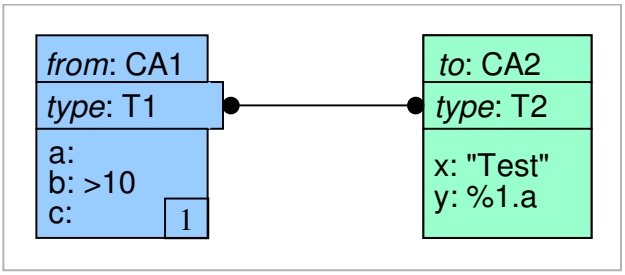

Figura 6 - Regra simples de política de colaboração (Fonte: Lima-Gomes, 2006)

A figura 7 apresenta dois exemplos de regras políticas com *Earliest* e *Latest*. No exemplo a esquerda, a regra política só é ativada quando ambos os eventos forem notificados. Sendo que o primeiro, que contém o *Predicate*, também precisa ser validado verdadeiro.

No exemplo à direita da figura 7, os componentes *Event* e *Trigger* estão agrupados por um *Earliest*. Quando um destes componentes for ativado, a política também será ativada. Regras de política como estas visam aguardar o atendimento de determinadas condições para serem ativadas (quanto ao estado da Supersessão). Entretanto, se um determinado evento é notificado antes que esta condição seja atendida, a regra também é ativada.

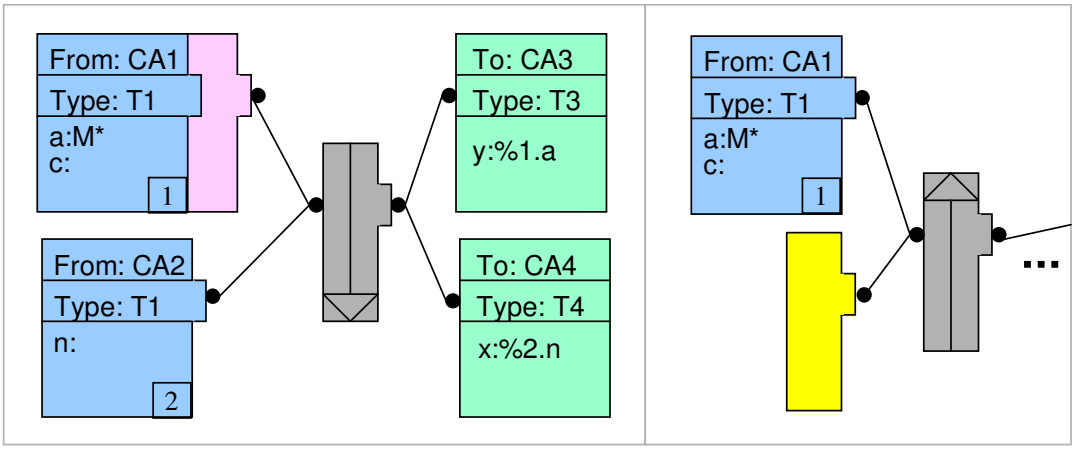

Figura 7 - Regra de política de colaboração utilizando o Earliest e Latest (Fonte: Lima-Gomes, 2006)

3.5 Considerações Finais

Neste capítulo foi visto o funcionamento e arquitetura de LEICA para a integração de aplicações colaborativas. Este ambiente foi escolhido como solução para integração do controlador de conferência proposto nesta dissertação com outras ferramentas colaborativas.

4 VOIP E O PBX IP ASTERISK

VoIP (*Voice over IP*) é o nome dado a transmissão de voz pela rede IP. Existe um grande interesse atualmente no uso da VoIP devido às várias vantagens desta tecnologia em termos de economia e integração com outros sistemas. Como conseqüência do crescimento na utilização de VoIP surgiram diversas aplicações de telefonia sendo implementadas como soluções abertas. O Asterisk [Asterisk, 2007] é uma delas, que implementa diversas funções encontradas em PBX convencionais.

Este capítulo descreve alguns aspectos da tecnologia VoIP e do servidor PBX IP Asterisk.

4.1 VoIP versus PSTN

O serviço de VoIP opera de maneira diferente do serviço fornecido por redes de telefonia tradicional, conhecidas como PSTNs (*Public Switched Telephone Network*). As redes PSTN usam conexões por comutação de circuito, onde é estabelecido um circuito dedicado de um telefone ao outro; a VoIP utiliza uma rede de comutação de pacotes, onde múltiplos dispositivos computacionais compartilham uma rede de dados.

Na telefonia por comutação de circuito é alocado um canal fixo de 64 kbps para cada chamada telefônica, fazendo uso do codec de voz G.711. Durante uma chamada telefônica, não se fala o tempo todo, em média é falado apenas 35% da duração da chamada. Mesmo que os usuários fiquem em silêncio, na telefonia PSTN são alocados recursos na rede para garantir os 64 kbps. Na VoIP, quando ocorre o silêncio, não se ocupa recursos da rede (não são gerados pacotes IP). Isto porque a maioria dos codecs possui detector de silêncio (*VAD – Voice Activity Detection*), resultando em um melhor aproveitamento da rede.

Além de explorar a supressão de silêncio e permitir o uso de codecs de voz que geram uma menor taxa de bits, a VoIP permite o uso mais eficiente de recursos de rede, pois pode-se explorar a multiplexação estatística das chamadas de voz. Por exemplo, na telefonia PSTN, 20 chamadas exigem 20 canais de 64 kbps (1280 kbps). Já na VoIP, graças à possibilidade de se explorar a supressão de silêncio, 20 chamadas de VoIP ocuparão uma taxa média inferior a 1280 kbps.

Uma das desvantagens da VoIP em relação à telefonia tradicional é a qualidade da voz. O serviço oferecido pelas redes IP geralmente é do tipo melhor esforço, que ao contrário da comutação por circuito não fornece garantias de qualidade, gerando um problema na implementação da VoIP. Para contornar isso, a VoIP pode fazer uso de outros codecs alem do G.711 como Speex, iLBC e GSM permitindo a redução da taxa de bits.

4.2 Chamadas VoIP

Uma chamada de VoIP compreende duas fases: a configuração de chamada e a conversação via voz em si. A configuração da chamada é o equivalente VoIP da obtenção do tom de discagem da telefonia, digitação do número do telefone, soar o telefone ou um sinal de ocupado, e tirar o telefone do gancho no outro lado.

Existem duas famílias de protocolos de configuração de chamadas [Soares, 2002]:

- Os protocolos H.323 [ITU-T, 1999] e MGCP (Media Gateway Control Protocol) [IETF, 2000] são oriundos da comunidade da telefonia. O H.323 é o protocolo mais usual e é na realidade uma família e padrões baseados em telefonia para multimídia, incluindo voz e videoconferência. MGCP é uma versão menos flexível, para uso em dispositivos mais baratos, como telefones domésticos.
- O SIP (*Session Initiation Protocol*) [IETF, 2002] é um protocolo desenvolvido pela IETF e é mais leve que o H.323. O SIP possui muitas características do HTTP e do SMTP. Do HTTP o SIP copiou o paradigma cliente-servidor e o uso de URLs e URIs. Do SMTP, o esquema de codificação de texto e o estilo do cabeçalho como *To*, *From*, *Date* e *Subject*

Em VoIP, a troca de dados de voz normalmente é feita através do protocolo RTP(Real-Time Transport Protocol) [Schulzrinne, 98], que é encapsulado em datagramas UDP e IP para transferência.

4.3 Componentes do sistema VoIP

Os componentes essenciais para um sistema de VoIP incluem: PBX IP, gateways VoIP e agentes usuários. Nas seções a seguir serão apresentados cada um destes componentes, detalhadamente.

4.3.1 Servidor de PBX IP e Gateways VoIP

Os servidores de telefonia PBX IP (*Private Branch Exchange Internet Protocol*) são geralmente baseados na arquitetura cliente-servidor, onde os serviços são solicitados e aguardado um resultado. O PBX IP tem o papel de servidor VoIP principal, assim como um servidor PBX convencional. Ele fornece funções e características semelhantes, como serviços de e-mail de voz, diretório, conferências, resposta interativa por voz (IVR), unidade de resposta audível (URA) e distribuidor automático de chamadas (DAC) e serviço de mensagem unificado (encaminhamento de mensagem de voz por e-mail).

A vantagem do uso de um servidor PBX IP é possuir os recursos normalmente encontrados em um servidor PBX e poder ser estendido de forma simples para outros fins. Um exemplo é a comunicação com servidor de e-mail para o serviço de mensagem unificado e personalização dos serviços de IVR e URA.

Os gateways VoIP têm a função de encaminhar datagramas de voz RTP para as redes IP. O gateway também tem a função de fazer a conversão de um codec para outro, como por exemplo, do G.711 para o iLBC. Outra função importante do gateway é a conexão entre redes IP e redes PSTN, sendo responsável pela comutação dos pacotes IP para transmissão de voz via circuitos virtuais.

4.3.2 Terminais VoIP

Os terminais VoIP, ou agente usuário na terminologia SIP, têm a função de ser o ponto terminal de recepção da VoIP. Ele deve ser capaz de estabelecer uma sessão de mídia com outro terminal. Isto pode ser feito através de software como softfones ou hardware como telefones IP.

4.4 Protocolo SIP

O SIP (*Session Initiation Protocol*) [IETF, 2002] é um protocolo de sinalização desenvolvido pela IETF para estabelecer conexões multimídia contendo áudio e/ou vídeo. O protocolo SIP é baseado em dois protocolos muito usados na Internet, que são os protocolos HTTP (HyperText Transfer Protocol) para navegação Web e o SMTP (Simple Mail Transport Protocol) para e-mails.

A Arquitetura SIP engloba basicamente agentes usuários, gateways SIP e servidores SIP. Os agentes usuários SIP são a combinação de dois componentes [Johnston, 2004]:

- Agente usuário cliente (UAC): Responsável por criar pedidos SIP do tipo: INVITE, ACK, OPTIONS, BYE, CANCEL e REGISTER - e enviá-los ao servidor. Em particular, a mensagem SIP INVITE permite realizar uma chamada VoIP.
- Agente usuário servidor (UAS): É responsável por receber os pedidos enviados pelo UAC. A comunicação entre UAC e UAS se dá pela arquitetura cliente-servidor.

O gateway SIP é uma aplicação responsável por fazer a interface da rede SIP para outro protocolo de sinalização. Como por exemplo, de SIP para H.323, ou ainda de SIP para a rede pública convencional PSTN.

Os servidores SIP têm a função de solicitar e atender pedidos SIP. Os servidores mais comuns são: *proxy*, *redirect* e *registrar*. O *proxy* tem a função de enviar o pedido da origem para o destino, sendo que o destino pode ser outro servidor *proxy*. O servidor *redirect* serve apenas para redirecionar o pedido do solicitante, sendo usado para quando o destino altera a sua localização. Por último o servidor *registrar,* que serve para registrar os clientes e averiguar de tempo em tempo a sua localização.

4.5 PBX IP Asterisk

O Asterisk [Asterisk, 2006] é um PBX IP muito utilizado principalmente por ser de código aberto GPL (*General Public License*) e oferecer diversas funcionalidades, como serviços de e-mail de voz, diretório, conferências, resposta interativa por voz (IVR),

unidade de resposta audível (URA) e distribuidor automático de chamadas (DAC). Ele pode rodar em diversas plataformas como Linux, BSD e MacOS X e suporta serviços como identificação de chamadas, ADSI, SIP, H.323 (como gateway e cliente) e MGCP (para gerenciamento).

Outros motivos que fazem o Asterisk ser muito utilizado são a redução de custos em comparação a soluções proprietárias, como também prover um ambiente de fácil desenvolvimento de aplicações. O Asterisk foi desenvolvido em linguagem C, permitindo que seu código possa ser modificado conforme as necessidades do sistema, além de aceitar outras linguagens de programação para efetuar soluções de telefonia [Gonçalves, 2006].

A maior limitação do Asterisk se encontra no desempenho da CPU usada como servidor para o serviço, onde é extremamente indicado o uso de uma máquina dedicada para o processamento de canais de voz, entre outras funcionalidades. Outra possibilidade é o seu agrupamento de servidores em cluster para fornecer aumento da escalabilidade.

4.5.1 Plano de Discagem do Asterisk

O Asterisk tem como núcleo de seu sistema o plano de discagem (*dialplan*), que define como devem ser tratadas as chamadas. O plano de discagem consiste de uma lista de instruções ou passos personalizáveis que o Asterisk deve seguir quando da ocorrência de chamadas de VoIP. O plano de discagem é composto de um ou mais contextos de extensões. Cada contexto de extensão é uma simples coleção de extensões usadas para implementar funcionalidades importantes, como segurança, roteamento e menus multi-níveis [Spencer, 2003]. No Asterisk, uma extensão é uma instrução que dispara uma ação, podendo ser associada a um ramal, interface, menu e outros.

Cada vez que é efetuada uma mudança no contexto ou principalmente nas extensões contidas no plano de discagem, é necessário reinicializar o servidor Asterisk. Para evitar esta reinicialização, pode-se utilizar a arquitetura ARA (*Asterisk RealTime Architecture*) [Meggelen, 2005]. Com esta arquitetura, o servidor Asterisk acessa um banco de dados que pode estar disponível no mesmo servidor Asterisk ou em outra máquina indicada em sua configuração para obter informações de contextos e de extensões do plano de discagem. Alguns dos bancos suportados pelo Asterisk são o PostgreSQL e o MySQL. Esta

arquitetura, quando habilitada, ainda permite acessar os contextos disponíveis nas extensões do próprio servidor Asterisk sem que haja conflitos entre as extensões e contextos existentes.

4.5.2 Gerenciamento de Conferências no Asterisk

No Asterisk, existem duas ferramentas principais disponíveis de forma gratuita que dão suporte à criação de conferências: *app_conference* [App_conference, 2006] e *Meetme* [Mahler, 2004].

A ferramenta *app_conference* é uma aplicação externa desenvolvida como uma alternativa ao serviço de conferência integrado ao Asterisk, o *Meetme*. Esta deve ser instalada a parte no servidor e tem características diferenciadas do *Meetme*. Ela destaca-se por não necessitar de uma fonte de relógio externa para fornecer controle de temporização nas conferências. Outro fator importante é seu desempenho em comparação ao *Meetme*, pois esta foi implementada para ser mais simples e mais escalável (suportar um número maior de conferências).

Diferente da *app_conference*, o *Meetme* requer uma fonte de tempo para o sincronismo das conferências, podendo ser feita por uma placa de telefonia ou por uma controladora USB. O *Meetme* possui mais recursos que o *app_conference*, como administração de conferências, criação dinâmica de conferências e senha para ingresso em conferências.

Tanto o *Meetme* quanto o *app_conference* possuem controle de palavra (floor control). O controle de palavra tem a função de permitir um participante da conferência ter o perfil de moderador, ouvinte, ou ainda permissão para falar. O moderador tem o direito de escutar e falar, cabendo a ele a possibilidade de excluir pessoas da conferência, limitar o número de participantes na conferência, modificar o papel dos usuários e incluir senha para o ingresso na mesma. Os papéis de ouvinte e locutor servem como permissões para os usuários finais, podendo ser alteradas apenas pelo moderador.

Outro fator importante das ferramentas existentes para conferências é o controle manual de inserção e retirada de participantes das conferências. Isto pode ser feito diretamente no terminal do Asterisk ou através de ferramentas externas como o Web-

MeetMe [Web-MeetMe, 2007] que faz o controle de palavra através de uma interface Web ou ainda por ferramentas de administração como o Asterisk Desktop Manager [Asterisk Desktop Manager, 2007].

4.5.3 Modelo de Conferências

De modo convencional, o servidor Asterisk dá suporte ao modelo de conferência *dial-in*. Neste caso, para um usuário entrar em uma conferência, ele precisa conhecer o número desta conferência e tomar a iniciativa de discar este número. O administrador pode incluir explicitamente no plano de discagem os números usados para conferências ou ele pode configurar o Asterisk para gerar dinamicamente uma conferência a partir da reserva de uma faixa de valores pré-determinada. Mas em ambos os casos, o usuário também deve conhecer o número usado pela conferência que deseja participar e realizar a discagem para a sua entrada na conferência. Utilizando a arquitetura ARA, podem ser criados números de conferências dinamicamente, sem a necessidade de reinicializar o Asterisk cada vez que é adicionada uma extensão. O funcionamento da ARA se dá no servidor da base de dados, utilizando consultas SQL. A base deve possuir uma tabela especialmente para os usuários e pode ser manipulada diretamente com um usuário válido para o banco de dados. Vale lembrar que o uso da ARA não desabilita o uso do plano de discagem normal do Asterisk, podendo ser executados simultaneamente.

4.5.4 Extensão do Servidor Asterisk

O Asterisk foi implementado de forma a permitir a realização de extensões em seus serviços e o controle de seu funcionamento implementado por aplicações externas ao servidor. A comunicação entre estas aplicações e o servidor Asterisk é feita via conexões TCP, e pode ser feita utilizando a *Fast AGI* (*Asterisk Gateway Interface*) ou usando a *Manager API*.

Em geral, usando o protocolo *Fast AGI*, as aplicações externas podem solicitar a execução de *scripts* contidos no plano de discagem do servidor Asterisk para serem executadas em uma segunda máquina, aliviando assim o uso de CPU do servidor primário.

Estes scripts podem ser implementados usando várias linguagens, como Perl, PHP e Python. Esta interface permite apenas o lançamento de scripts AGI para serem executados onde a aplicação externa não receberá nenhum resultado posterior pelo Asterisk caso isto não seja implementado.

A *Manager API* [Manager API, 2006] permite, além de solicitar a execução de funções no Asterisk, observar eventos ocorridos no interior do servidor Asterisk. Três conceitos básicos de funcionamento são definidos na *Manager API*: ações, respostas e eventos. As ações são comandos requisitando que uma operação em particular seja executada no servidor, e em seguida uma resposta é encaminhada ao solicitante apresentando os resultados da ação. Os eventos são informações geradas pelo Asterisk durante a sua execução.

4.5.5 Asterisk-Java

Como visto na seção anterior, o Asterisk pode ser manipulado pela Manager API e Fast AGI para mudanças em seu estado. No caso da adoção da linguagem Java para a implementação de aplicativos para o Asterisk, pode-se usar o pacote Asterisk-Java [Asterisk-Java, 2007]. Este pacote consiste de um conjunto de classes Java que facilitam a interação tanto da *Fast AGI* quanto da *Manager API*.

O Asterisk-Java permite que aplicações em Java possam ser desenvolvidas para o servidor Asterisk. Normalmente as aplicações fazem uso do Manager SIP, que faz a autenticação no servidor e possibilita executar funções como efetuar ações e receber eventos. No caso do Fast AGI, o Asterisk-Java disponibiliza métodos para a construção de scripts que podem ser iniciados no Asterisk.

4.6 Considerações Finais

Este capítulo teve como objetivo apresentar alguns conceitos de VoIP e introduzir o funcionamento do servidor Asterisk. O Asterisk apresenta várias soluções para VoIP e conferência, mas não oferece um mecanismo de integração automática com as outras ferramentas de um mesmo ambiente colaborativo. Os usuários no ambiente em questão

teriam que ingressar manualmente em conferências ou utilizar ferramentas como a citada Web-MeetMe para convidar os participantes. Isto gera uma limitação no uso de grupos de trabalho dinâmicos. No próximo capítulo será apresentado o controlador de conferências para ambientes de trabalho colaborativo.

5 CONTROLADOR DE CONFERÊNCIAS PARA AMBIENTES DE TRABALHO COLABORATIVO

A conferência é um serviço fundamental que deve estar presente em ambientes colaborativos, pois permite a comunicação direta entre seus participantes com áudio e opcionalmente vídeo. Como visto nos capítulos anteriores, ferramentas de conferência que não são integradas com outras ferramentas colaborativas obrigam os membros a gerenciar manualmente sua entrada e saída em conferências apropriadas. Em sessões onde todos os membros devem estar na mesma conferência, esta operação manual não perturba o trabalho colaborativo, mas dificulta o processo quando devem dinamicamente criar e migrar para grupos de trabalhos individuais (requerendo a existência de uma conferência para cada grupo de trabalho).

Neste capítulo é proposto o Controlador de Conferências que pode ser facilmente integrado a outras ferramentas colaborativas. O termo Controlador de Conferências é adotado neste trabalho para referenciar uma entidade capaz de realizar o gerenciamento do serviço de conferência. Conforme visto na seção 2.4, um Controlador 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 o controle de palavra (*floor control*).

Para facilitar a integração do Controlador de Conferências a outras aplicações colaborativas, foi adotado o ambiente LEICA (*Loosely-coupled Environment for Integrating Collaborative Applications*) [Lima-Gomes, 2005]. Como visto no capítulo 3, LEICA permite uma integração fracamente acoplada de aplicações colaborativas possibilitando a abstração dos detalhes internos de cada aplicação, facilitando assim o processo de integração. Apoiando-se no ambiente LEICA para implementar a integração do Controlador de Conferências a outras aplicações colaborativas, 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.

O Controlador de Conferências proposto gerencia áudio-conferências que são realizadas utilizando um PBX IP Asterisk [Asterisk, 2007]. A vantagem do Asterisk se dá principalmente ao seu gerenciamento eficiente de ligações e suporte a outras aplicações externas para adicionar novas funcionalidades ao PBX.

O Controlador de Conferências foi implementado na forma de uma aplicação Java (*standalone*). Esta linguagem foi escolhida para facilitar a integração ao ambiente LEICA, como também pela existência de um conjunto de pacotes Java que simplificam a comunicação do controlador com o servidor Asterisk.

5.1 Escopo do Trabalho

A presente dissertação não propõe um Controlador de Conferências genérico que atenda todos os requisitos levantados na seção 2.4, e sim um controlador permitindo a inclusão de serviços de conferência em ambientes de trabalho colaborativos.

Uma simplificação adotada refere-se aos modelos de conferências suportados. O Controlador de Conferências suporta apenas conferências *dial-out*, pois o gerenciamento de usuários é realizado automaticamente pelo controlador via troca de mensagens com a aplicação colaborativa integrada. Neste caso, o usuário receberá chamadas para participar de conferências. Isto simplifica o gerenciamento do trabalho colaborativo, pois não requer que os usuários controlem o tempo de acesso à conferência, nem a identificação de que conferência devem participar.

5.2 Visão Geral da Arquitetura do Controlador de Conferências

A figura 8 apresenta os principais componentes envolvidos com a operação do Controlador de Conferências proposto. Os componentes são:

- **Usuários**: Cada usuário que interage com a aplicação colaborativa recebe automaticamente chamadas VoIP para participar de conferências via seu agente usuário SIP (softfone, telefone IP etc.). De acordo com a necessidade da aplicação colaborativa, o Controlador de Conferências cria dinamicamente conferências e

convida os usuários a se juntarem a esta, também sendo capaz de expulsá-los da conferência, caso necessário.

- **Servidores de aplicações colaborativas**: servidores que, via o ambiente LEICA, podem ser integrados ao Controlador de Conferências.
- ***Wrapper* LEICA**: Gerado e associado ao Controlador de Conferências de maneira a integrar este último ao ambiente LEICA.
- ***API* do Controlador de Conferências**: API pela qual o *Wrapper* LEICA é capaz de controlar a operação do Controlador de Conferências e receber notificações de eventos ocorridos internamente ao Asterisk.
- **Módulos do Controlador de Conferências**: Os gerenciadores de conferências, de usuários, de sessão e do Asterisk interagem com o servidor Asterisk (via o pacote Asterisk-Java)que implementam as funcionalidades oferecidas pela API do Controlador de Conferências. Basicamente, estes módulos são capazes de solicitar ao PBX Asterisk a execução de determinadas ações (*e.g.* desconectar um usuário de uma conferência), e de capturar respostas e eventos ao nível do PBX Asterisk (*e.g.* o estabelecimento de uma nova conferência).
- **Pacote Asterisk-Java**: Pacote Java permitindo aos módulos do Controlador de Conferências interagirem com o PBX Asterisk.

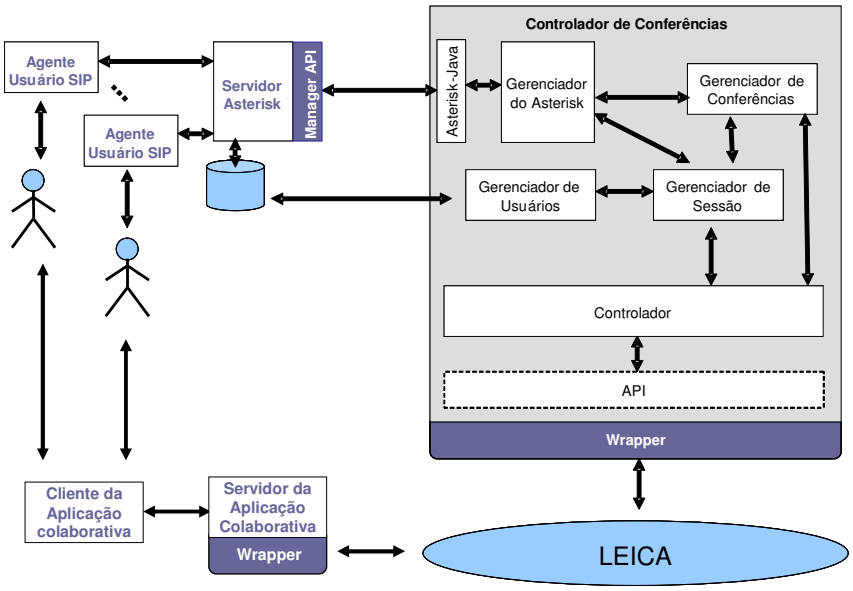

Figura 8 - Arquitetura do Controlador de Conferências

Como descrito em [Siggelkow 2003], a divisão de tarefas entre diferentes grupos de trabalho durante a realização de um trabalho colaborativo é uma prática comum visando a otimização deste trabalho. Torna-se, portanto, importante que um sistema de conferências voltado a ambientes colaborativos 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.

A figura 9 apresenta o diagrama de classe do controlador de conferências. As seções que seguem detalham os componentes do sistema Controlador de Conferências.

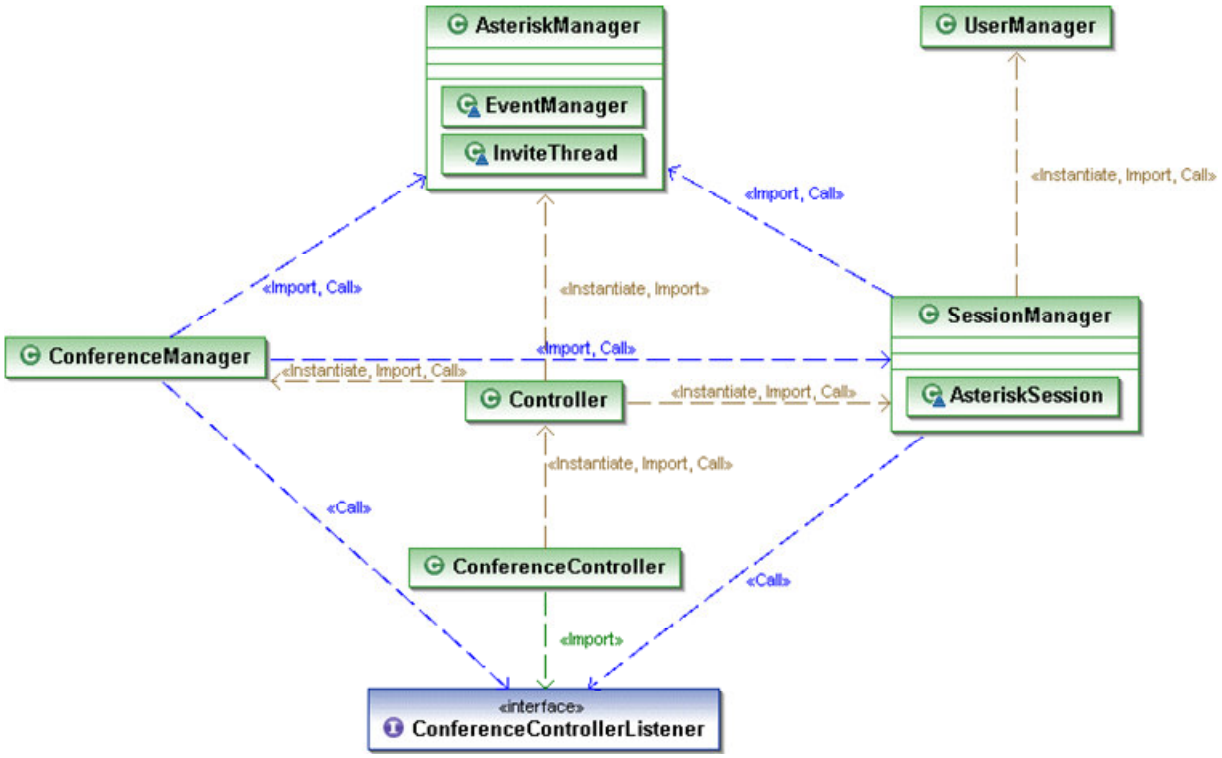

Figura 9 - Diagrama de classe do Controlador de Conferências

5.3 Configuração do PBX Asterisk

Como descrito no capítulo 3, é necessário o uso de uma aplicação de conferências para Asterisk, onde existem duas opções: *Meetme* e *app_conference*. Neste trabalho foi adotada a aplicação *Meetme*, por possuir mais recursos de ações e eventos em comparação a atual implementação do *app_conference*. O *Meetme* permite criar e gerenciar conferências através de parâmetros passados pelo controlador para o Asterisk, como emudecer participantes, anunciar chegada de participantes e retirar participantes.

Para que participantes fossem adicionados de forma dinâmica nas conferências, foi adotada ARA (*Asterisk Realtime Architecture*) [Meggelen, 2005]. A ARA permite adicionar e retirar usuários em tempo real do servidor, através de um banco de dados, que pode estar em um servidor isolado ou no próprio host do servidor Asterisk. Para a proposta foi adotado a base de dados MySQL e instalada no mesmo host do servidor Asterisk.

5.4 API do Controlador de Conferências

Esta API define uma classe oferecendo métodos para acesso a funcionalidades do Controlador de Conferências e uma interface oferecendo mecanismos para receber notificações de eventos ocorridos. A figura 10 apresenta o diagrama de classes da API do controlador de conferências.

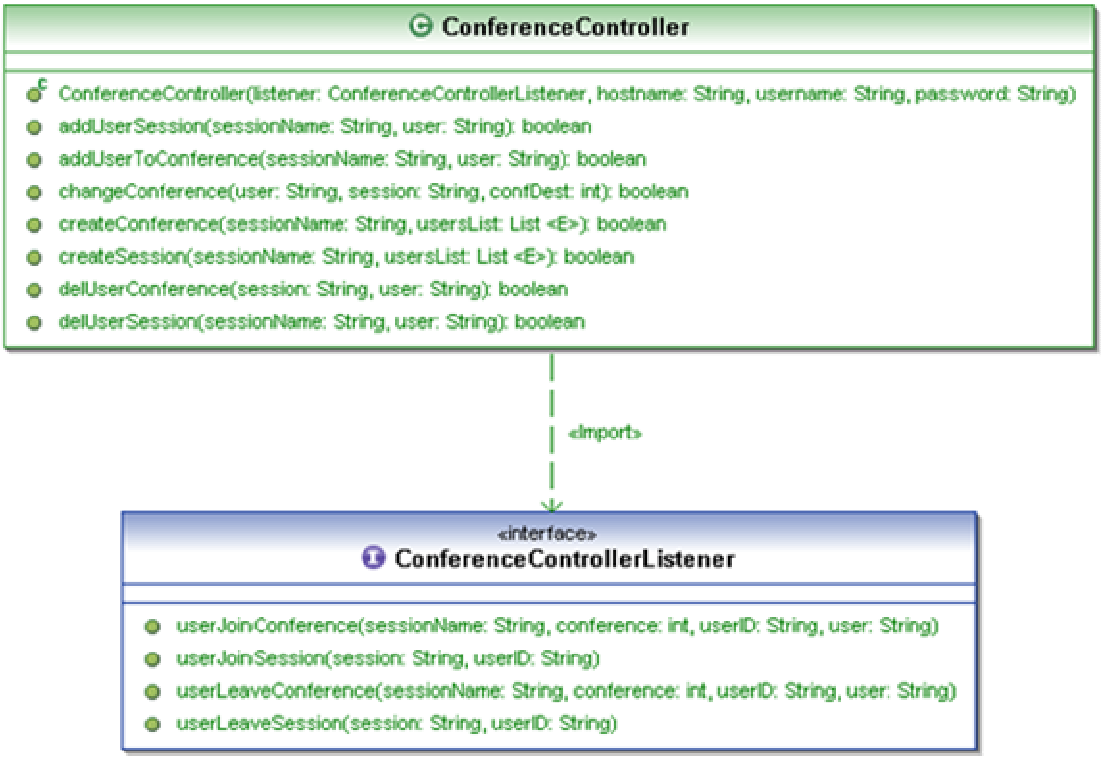

Figura 10 - Diagrama de classe da API do Controlador de Conferências

5.4.1 Interface ConferenceControllerListener

Uma aplicação cliente usando a API do Controlador de Conferências deve criar uma classe que implementa a interface ConferenceControllerListener. Um objeto desta classe poderá receber a notificação de quatro eventos de conferência recebidos do servidor Asterisk:

- userJoinSession: Este evento ocorre quando há um ingresso de usuário em uma sessão. Esta classe tem como parâmetros o nome da sessão e a identificação do usuário.
- userLeaveSession: Este evento ocorre quando da saída de um participante de uma sessão específica. Os parâmetros são os mesmo do evento userJoinSession.

- userJoinConference: Este evento ocorre quando da entrada de um membro da sessão em uma conferência. Os parâmetros informados são: nome da sessão, número da conferência, identificação do usuário e seu nome.
- userLeaveConference: Este evento ocorre quando da saída de um participante de uma conferência específica. Os parâmetros são os mesmo do evento userJoinConference .

5.4.2 Classe ConferenceController

O objeto desta classe realiza o gerenciamento do Controlador de Conferências, implementando todos os métodos oferecidos na API. Os métodos que compõem esta classe são:

- addUserSession: Adiciona um usuário em uma sessão específica;
- addUserToConference: Permite adicionar um ou vários usuários em uma conferência específica;
- changeConference: realiza a troca de conferência de um usuário para outra;
- createConference: Cria conferências em uma sessão. Não foi necessário um método para destruição da conferência; pois ao sair o último participante o Asterisk já a dá como encerrada;
- createSession: Cria uma sessão, passando uma lista de usuários para ingresso. O método addUserSession pode ser usado posteriormente para incluir mais membros na sessão;
- delUserConference: Expulsa o usuário definido da conferência;
- delUserSession: Retira o usuário da sessão, sendo que se este tiver em uma conferência ocorrerá também a sua expulsão .

5.5 *Wrapper* LEICA

Para a integração do Controlador de Conferências à LEICA, foi seguido o procedimento de integração definido por este ambiente (apresentado na seção 3). Foram

criados os seguintes arquivos: Arquivo de Dados Específicos e o Arquivo de Atributos e API.

O Arquivo de Dados Específicos descreve as informações que o Controlador necessita para a criação de uma nova sessão. Este arquivo define a LEICA uma lista de usuários (*userList*) para abrir uma sessão. A Figura 11 ilustra o Arquivo de Dados Específicos.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.colab.org"
            xmlns=" http://www.colab.org "
            elementFormDefault="qualified">
 <xsd:element name="spSspecs">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="usersList" minOccurs="1" maxOccurs="1" >
     <xsd:annotation>
       <xsd:documentation>Each tag userList contains a list of tags user </xsd:documentation>
     </xsd:annotation>
     <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="user" minOccurs="1" maxOccurs="unbounded" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Each tag user contains a username </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

Figura 11 - Arquivo de Dados Específicos

O Arquivo de Atributos e API descreve os tipos de acontecimentos/ações que podem ser notificados/executados através da API do Controlador de Conferências. Na seqüência, é utilizado o módulo *API Factory* para gerar o *Wrapper* adaptado ao Controlador de Conferências. A figura 12 apresenta um exemplo com alguns métodos da API no Arquivo de Atributos e API. Este arquivo apresenta dois eventos que são o de notificação de entrada e a notificação de saída de conferências. A seção de ações apresenta: criar conferências, mudar o usuário da conferência, expulsar usuário, conectar e desconectar. As ações de conectar e desconectar são interpretadas no LEICA para adicionar e retirar membros da sessão.

```
<CA id="Asterisk" wsRouter="http://malva.inf.ufsc.br:8090/soap/servlet/rpcrouter"
 uddiInqURL="http://malva.inf.ufsc.br:8080/juddi/inquiry"
 uddiPubURL="http://malva.inf.ufsc.br:8080/juddi/publish" >
 <attributes>
  <name> Asterisk </name>
  <description> IP PBX </description>
  <type> AUDIOCONFERENCE </type>
  <maxRolePerUser> 1 </maxRolePerUser>
  <distribution> client-server </distribution>
  <userApplication>
  </userApplication>
 </attributes>
 <API>
  <evtsAPI>
   <event type="userJoinedConference">
    <description> Event - Find out to which session this conference belongs to</description>
    <param name="conferenceNum" valueType="integer"/>
    <param name="userID" valueType="string"/>
    <param name="user" valueType="string"/>
   </event>
   <event type="userLeaveConference">
    <description> Event - Find out to which session this conference belongs to</description>
    <param name="conferenceNum" valueType="integer"/>
    <param name="userID" valueType="string"/>
    <param name="user" valueType="string"/>
   </event>
  </evtsAPI>
  <actsAPI>
   <action type="createConference">
    <description> Action - Add 1 user only for the conference</description>
    <param name="usersList">
     <description> The list of users to connect to the conference </description>
     <sequence base="string" size="*"/>
    </param>
   </action>
   <action type="changeConference">
    <description> Change user from a conference to another and change the session if necessary</description>
    <param name="user" valueType="string"/>
    <param name="confSource" valueType="integer"/>
    <param name="confDest" valueType="integer"/>
   </action>
   <action type="delUserConference">
    <description>Remove user from a conference</description>
    <param name="user" valueType="string"/>
   </action>
   <action type="CONNECT" >
    <description> In the implementation, this action will be translated into a call to addUserSession</description>
    <param name="user" valueType="string">
     <description> Corresponde ao Uid </description>
    </param>
   </action>
   <action type="DISCONNECT" >
    <description> In the implementation, this action will be translated into a call to delUserSession</description>
    <param name="user" valueType="string">
     <description> Corresponde ao Uid </description>
    </param>
   </action>
  </actsAPI>
  <gestionAPI>
  </gestionAPI>
 </API>
</CA>
```

Figura 12 - Arquivo de Atributos e API

A última etapa da geração do *Wrapper* para o Controlador de Conferências é a implementação da classe *ApplicationInterface*. Esta classe permite o acoplamento da API do Controlador de Conferências com o *Wrapper*, fazendo a ligação entre os métodos correspondentes dessas duas classes.

5.6 Controlador

Este módulo, cujo diagrama de classe é apresentado na figura 13, é responsável pelo gerenciamento do controlador de conferências. Na instanciação de um objeto ConferenceController também é instanciado um objeto da classe Controller. Este último instancia os módulos ConferenceManager, SessionManager, AsteriskManager. Ele também implementa todas as funcionalidades oferecidas pela API do Controlador de Conferências.

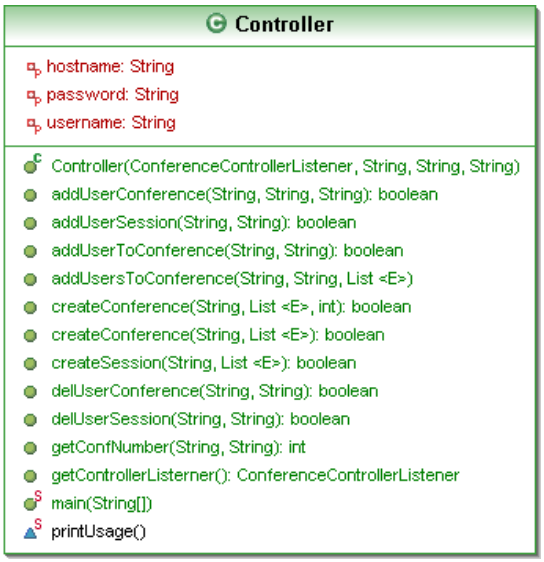

Figura 13 - Diagrama de classe do controlador

5.7 Gerenciador de Conferências

O Gerenciador de Conferências, cujo diagrama de classe é apresentado na figura 14, é responsável por gerenciar as conferências durante um trabalho colaborativo. Este gerenciador oferece as seguintes funcionalidades:

- Criação de conferências (createConference) assim como incluir e excluir participantes das conferências (addUserToConference, addUsersToConference, delUserConference). Na operação de criar conferência, é informado o nome da sessão na qual a conferência deve ser criada, a lista de participantes e opcionalmente o número da conferência (número utilizado para criar a conferência no PBX Asterisk). Caso o número

da conferência não seja informado, o gerenciador gera um identificador único. O número da conferência (informado ou gerado aleatoriamente) sempre é verificado para saber se está em uso por outra conferência. Este gerenciador também faz uso do gerenciador do Asterisk para efetuar os métodos de convite (inviteUser) e expulsão (kickUser) quando necessários.

- Adição e remoção de usuários na conferência (addUserConference, addUserToConference, delUserConference). Os três métodos têm interação com o gerenciador do Asterisk para efetuar as chamadas. Os métodos de adição podem incluir participantes em uma conferência especificada ou ainda adicionar em uma onde se encontra outro participante.

Figura 14 - Diagrama de classes do Gerenciador de Conferências

Durante a criação de uma conferência, este módulo interage com o Gerenciador de Sessão notificando quais usuários da sessão fazem parte da conferência recém criada.

Os participantes de conferência, diferente dos usuários de sessão, precisam ser registrados junto ao servidor Asterisk para que possam fazer sua autenticação de serviço VoIP. Antes de conectar um usuário a uma conferência, o Asterisk faz uma consulta na base de dados para verificar se o mesmo encontra-se cadastrado, para então efetuar sua entrada na sala de conferência.

Finalmente, o Gerenciador de Conferências solicita a criação da conferência junto ao servidor Asterisk, solicitando as chamadas VoIP para os participantes desta.

5.8 Gerenciador de Sessão

O conceito sessão foi introduzido no Controlador de Conferências de maneira a facilitar a integração com outras aplicações colaborativas, via o ambiente LEICA. Neste controlador, uma sessão é definida por um conjunto de participantes da sessão e um conjunto de conferências associadas à sessão. Em uma sessão, cada conferência é identificada por um número e o subconjunto de participantes da sessão que integram a conferência. Note que este conceito foi implantado no Controlador de Conferências, não existindo no PBX Asterisk.

Durante a realização de um trabalho colaborativo, uma sessão representa um grupo de participantes (os membros da sessão) interagindo via uma ou mais ferramentas de colaboração, visando uma meta comum. Estes participantes podem se organizar dinamicamente em grupos de trabalho para realizar ações específicas. Desta forma, o Controlador de Conferências possibilita a divisão do grupo em diferentes conferências (formando grupos de trabalho).

O Gerenciador de Sessão, cujo diagrama de classes é apresentado na figura 15, é responsável por controlar a entrada e saída de usuários das sessões e sua relação com as conferências existentes. Para isto, ele mantém os registros das sessões criadas. As principais funções oferecidas por este gerenciador são:

- Criação e de Sessões (createSession): Todas as sessões devem possuir um nome único para que sejam registradas no gerenciador. Na criação de uma sessão (método createSession), é informado o seu nome e a lista de membros.
- Adição e remoção de usuários na sessão (addUserSession, delUserSession): Cada membro da sessão possui um nome de usuário único que é igual ao identificador SIP para as ligações VoIP no servidor Asterisk. Estes membros podem participar de várias sessões ao mesmo tempo, mas não podem participar mais de uma vez da mesma sessão.
- Inserção e remoção nas tabelas hash (addUser, addUserConference, removeConference, removeUser): Os métodos restantes servem para que haja

um controle dos usuários dentro do controlador. Isto é feito através de tabelas hash que guardam informações das sessões, conferências e usuários.

Figura 15 - Diagrama de classe do Gerenciador de Sessão

O gerenciador mantém as informações das sessões atuais que são mantidas na forma de um objeto da classe AsteriskSession, todos incluídos em uma hashtable. Os objetos da classe AsteriskSession mantém uma série de informações sobre a sessão: o nome da sessão (*sessionName*), usuários da conferência (usersConf), identificação do usuário no servidor Asterisk (usersID), usuários da sessão (usersSession).

5.9 Gerenciador de Usuário

O Gerenciador de Usuários, cujo diagrama de classe é apresentado na figura 16, oferece uma série de métodos para inserir e coletar informações sobre os usuários através da base de bancos do servidor Asterisk. Os métodos oferecidos são:

- Criação e remoção de usuários (createUser, delUser): Permite a criação de usuários SIP para autenticação no servidor Asterisk. Como visto na configuração do servidor, o Asterisk utiliza uma base de dados MySQL para funcionar em tempo real. Os comandos de criação e remoção são feitos através de consultas diretamente na base de dados.
- Leitura de informações sobre o usuário (getInfoUser, userExists): O gerenciador possui acesso à base de dados do servidor Asterisk, permitindo obter informações dos participantes das conferências como por exemplo, o codificador de áudio utilizado, caixa postal de voz, e endereço IP de origem atual.

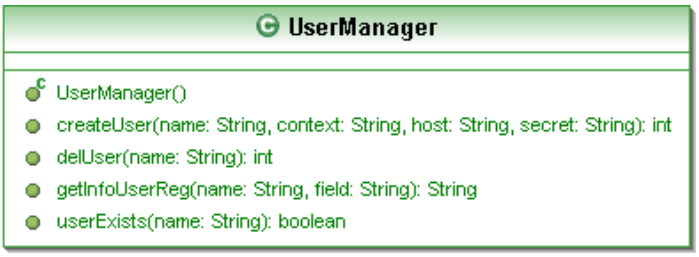

Figura 16 - Diagrama de classe do Gerenciador de Usuário

5.10 Gerenciador do Asterisk

Este módulo é responsável pela comunicação com o servidor Asterisk, através do Asterisk-Java. Além disso, ele é utilizado para capturar determinados eventos internos ao servidor Asterisk.

A figura 17 apresenta o diagrama de classe deste módulo. A seguir os principais métodos deste módulo são apresentados:

- startEventManager: Método que tem como função iniciar a thread do gerenciador de eventos. Esta thread possibilita ao gerenciador do Asterisk receber notificações de eventos que ocorrem dentro do servidor Asterisk (via pacote Asterisk-Java) e notificar estes ao Controlador de Conferência.
- inviteUser: Onde é feito o convite de usuário para o servidor Asterisk, passando o usuário e o número da conferência de ingresso.

- inviteUserThread: É uma thread definida que utiliza o inviteUser para permitir o convite simultâneo (em paralelo) de vários usuários.
- kickUser: Método que solicita ao Asterisk que retire o usuário da conferência especificada.
- infoUser: Coleta informações do usuário no Asterisk, como tempo de permanência e em qual conferência este se encontra no momento.
- countUsers: Mostra o número de participantes que estão na conferência no momento.
- waitEventManager: Método utilizado junto ao startEventsManager para aguardar que certos eventos ocorram antes que outros métodos sejam chamados.

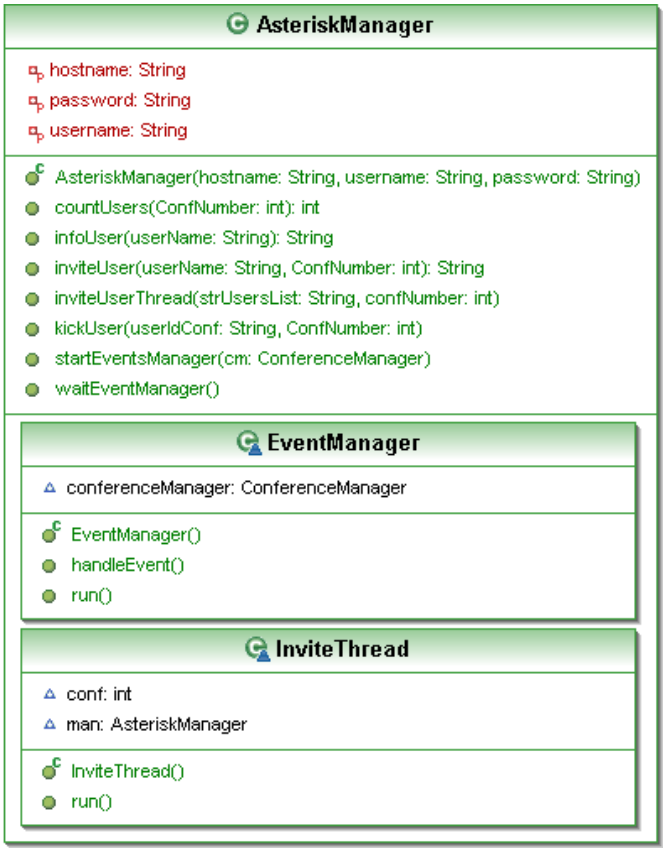

Figura 17 - Diagrama de classe do Gerenciador do Asterisk

O Notificador de Eventos (classe EventManager) é responsável por notificar eventos ocorridos no servidor Asterisk e também eventos relacionados às sessões para notificar

aplicações externas, através da API do controlador. Os eventos que podem ser enviados pelo Asterisk à API são:

- userJoinConference: Evento chamado quando um usuário se conecta a uma conferência. Este evento informa a conferência onde o usuário foi alocado, o identificador único do usuário e o nome do usuário. Na chamada deste evento tambem ocorre a inserção do usuário em uma lista de participantes para controle da API.
- userLeaveConference: Este evento informa a saída de um participante da conferência. Como acontece no evento de entrada este também informa a conferência, a identificação do usuário e o nome do usuário. Ocorre neste evento também a retirada do usuário da lista de conferência correspondente da API.
- MeetMeTalkingEvent: Este evento informa à API quando um usuário começa a falar. Este evento pode informar a identificação do participante, a conferência e o nome do participante.
- MeetMeStopTalkingEvent: Como o evento anterior, este agora informa quando um usuário terminou de falar em uma conferência. Os dados passados à API são os mesmos: a identificação do participante, a conferência e o nome do usuário.

Quando o controlador efetua ações diretamente no servidor Asterisk, ele também fica responsável por enviar as respectivas notificações a aplicações externas, no contexto desta proposta LEICA.

5.11 Considerações Finais

Neste capítulo foi apresentada a estrutura e o funcionamento do Controlador de Conferência junto ao servidor Asterisk. A API do controlador pode fornecer diferentes métodos para que se possam criar conferências e como gerenciar os usuários destas. Além desta função, outra importante é gerencia e o controle de sessões como também os membros destas. Junto a LEICA o controlador pode ser integrado a outras aplicações que estejam em funcionamento no ambiente, proporcionando uma forma de comunicação

automática aos participantes, sem que estes precisem se preocupar na troca de conferências. Comparada a outras ferramentas existentes para conferência, o Controlador tem a vantagem desta integração que permite a automação de vários processos não encontrado em ambientes colaborativos atuais.

No capítulo seguinte será abordada a integração do controlador junto ao navegador CoLab para formação de um ambiente colaborativo de teste. Esta integração servirá de validação do controlador de conferências proposto.

6 CO-NAVEGAÇÃO COM SUPORTE A CONFERÊNCIA

Conforme apresentado na seção 2.3, as aplicações de co-navegação permitem a um grupo de usuários navegarem pela Web em conjunto, cada um em seu próprio computador. 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.

A ferramenta de co-navegação CoLab (seção 2.5.1) permite a navegação colaborativa, mas não oferece nenhum mecanismo de comunicação entre os participantes da sessão de co-navegação. Desta forma, CoLab deveria ser integrada com outra ferramenta de comunicação, ex. a um serviço de conferência, para viabilizar a realização de sessões de co-navegação.

Este capítulo apresenta um estudo de caso de integração do controlador de conferências, proposto nesta dissertação e a ferramenta de co-navegação CoLab, fazendo uso de LEICA como ambiente de integração. O objetivo é criar um sistema de co-navegação com suporte a conferências. 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 meta deste estudo de caso é validar as funcionalidades do Controlador de Conferências proposto, deixando que as conferências sejam realizadas de forma automática com a sincronização e desincronização dos usuários durante uma sessão de co-navegação.

6.1 Ambiente de Co-navegação

A Figura 18 apresenta os principais componentes do ambiente de co-navegação implementado:

- Serviço de Configuração de Sessão (SCS) – O SCS de LEICA é usado para criar e iniciar supersessões. Cria-se o arquivo de configuração da Supersessão que se chama “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 ingressar em uma Supersessão, o usuário precisa executar a aplicação LClient. O LClient inicialmente se conecta ao SCS e recebe as informações de configuração para fornecer a Supersessão (escolhida pelo usuário). Baseando-se na Política de Colaboração, LClient inicia o cliente CoLab (carregado em um navegador Web contendo a página 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 o CoLab como o Controlador de Conferências notificam eventos durante a Supersessão (por exemplo, indicando que o usuário A sincronizou com o 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 um áudio-conferência conectando os usuários A e B quando estes se juntarem ao mesmo grupo de trabalho CoLab).

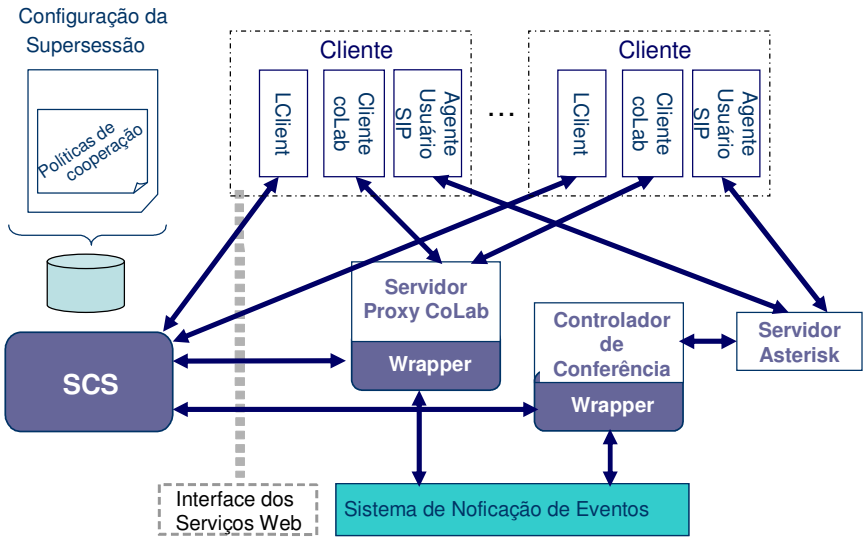

Figura 18 - Ambiente de Co-navegação

6.2 Exemplo Ilustrativo de Operação

A fim de ilustrar o uso do 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 uma 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 19 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 a D e E.

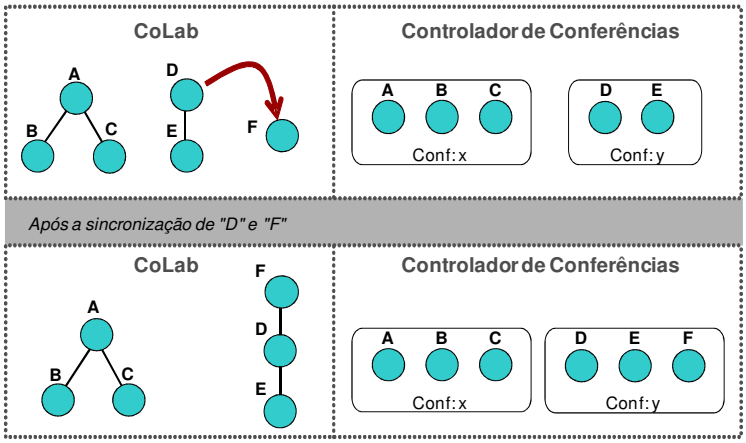

Figura 19 - Cenário de Co-navegação

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 Supersessão. Para que outros tipos de comportamento sejam especificados, basta alterar as regras dessa política. Por exemplo, (i) pode-se manter um usuário em uma 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.

6.3 Cenário de Integração

O ambiente de execução para a validação dos testes foi composto pelo cenário ilustrado na figura 20. O ambiente contou com dois servidores (malva e boldo) e duas máquinas clientes, atuando como terminais dos usuários finais (carqueja e cidreira). Todos estes equipamentos fazem parte de uma mesma rede local. Um dos servidores (boldo) continha apenas o Proxy CoLab enquanto o outro (malva) continha o servidor Asterisk, o

Controlador de Conferências e LEICA (SCS e UDDI). Nas máquinas clientes foi necessário instalar um softfone para as conferências e também uma cópia do LClient (LEICA Client).

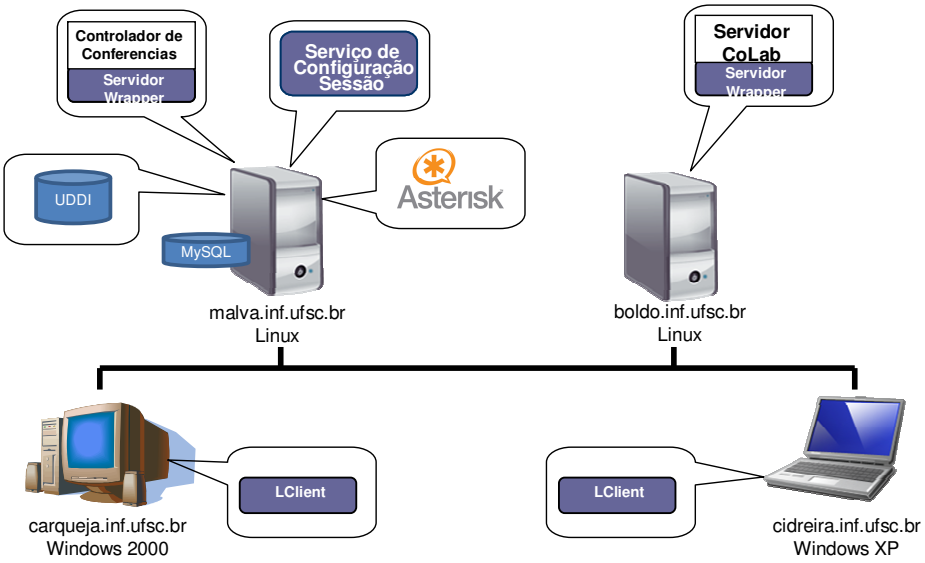

Figura 20 - Cenário de Co-navegação com suporte à conferência

6.4 Processo de Integração CoLab e Controlador de Conferências

O primeiro passo realizado para a integração de CoLab com o Controlador de Conferências foi definir a semântica desta integração, que é definida via políticas de colaboração de LEICA. Existem diversas possibilidades em termos da semântica de integração destas duas ferramentas. A figura 21 apresenta duas possíveis políticas de colaboração. A política ilustrada na figura 21-A define um cenário simples de integração, onde todos os membros de uma sessão CoLab se juntam a uma mesma conferência. Isto implica que todos os usuários do sistema de co-navegação estarão na mesma conferência, independente da forma como eles se sincronizam na ferramenta CoLab. Mesmo que estes usuários estejam em diferentes grupos de trabalho, eles estarão na mesma conferência.

A política especificada na figura 21-B define um cenário de integração mais complexo, onde os membros de cada grupo de trabalho do CoLab (criados dinamicamente) estarão em uma conferência separada (ex. uma conferência para cada grupo de trabalho CoLab). No contexto de uma sessão CoLab, podem ser formados diferentes grupos de trabalho durante o trabalho colaborativo, assim cada grupo precisa ser associado a uma

conferência diferente. Conseqüentemente, quando grupos de trabalho da sessão CoLab são criados e destruídos através de sincronização, as respectivas conferências precisam ser criadas e destruídas pelo Controlador de Conferência. A regra 1 especifica que quando o usuário se conecta a Supersessão, ele será incluído em uma sessão CoLab e em uma sessão no Controlador. A regra 2 define que quando um evento de sincronização ocorrer em CoLab, implica a adição do usuário escravo da sincronização (synced) ser alocado na mesma conferência do usuário mestre. Finalmente a regra 3 define que a liberação de sincronização no CoLab implica na criação de uma nova conferência onde será inserido o usuário saindo da sincronização (synced), e a conseqüente retirada deste da conferência anterior .

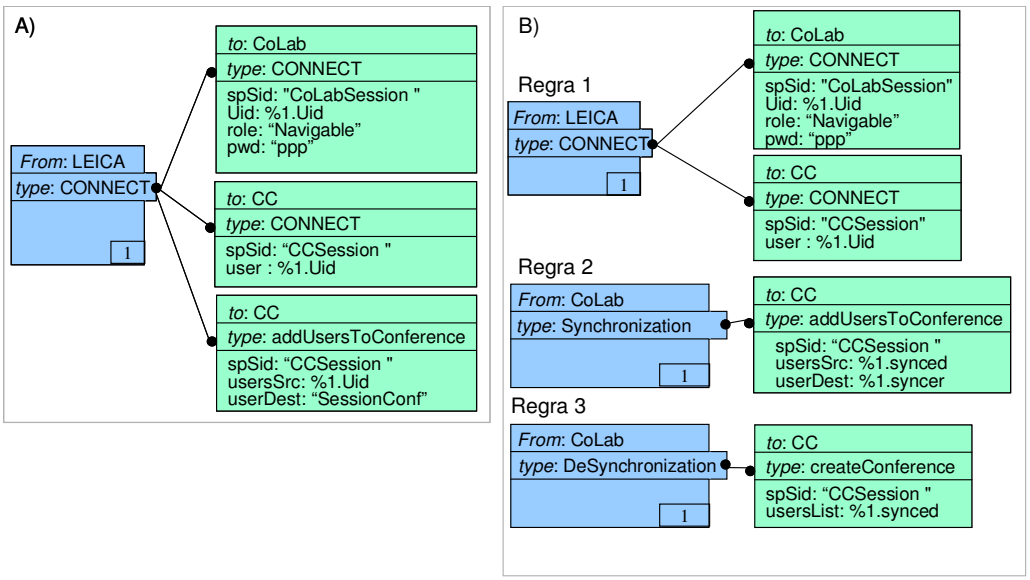

Figura 21 - Exemplo de políticas de colaboração

Além disto, as políticas de colaboração poderiam por exemplo incluir controle de palavra, associando privilégios a diferentes papéis de usuários. A política poderia definir que os usuários assíncronos do CoLab (raiz dos nós SDTs) poderiam falar e escutar enquanto os outros usuários poderiam apenas escutar.

6.5 Teste das Funcionalidades

A fim de testar o bom funcionamento do controlador de conferências, foi simulado um cenário de trabalho colaborativo. O cenário de operação do sistema de co-navegação

testado foi composto de três usuários (A, B e C) fazendo uso do CoLab junto a um softfone. O usuário A solicitou um comando “You Follow Me” no co-navegador para que o usuário B sincronizasse a ele. Quando ocorreu o aceite deste usuário, automaticamente o Controlador gerou um convite para o softfone do usuário A. Apenas quando o usuário atende a chamada é que ocorre o convite para entrada na conferência, finalizando o processo de sincronização da parte do Controlador. Na seqüência, o usuário C pede a sincronização com B através de um comando “I Follow You”, lembrando que o usuário B está sincronizado A. Quando o A desiste da sincronização em CoLab com o comando “I Leave”, o Controlador envia um convite para este ingressar em outra conferência, retirando este participante da conferência anterior, deixando os usuários B e C. A figura 22 apresenta os passos do cenário descrito.

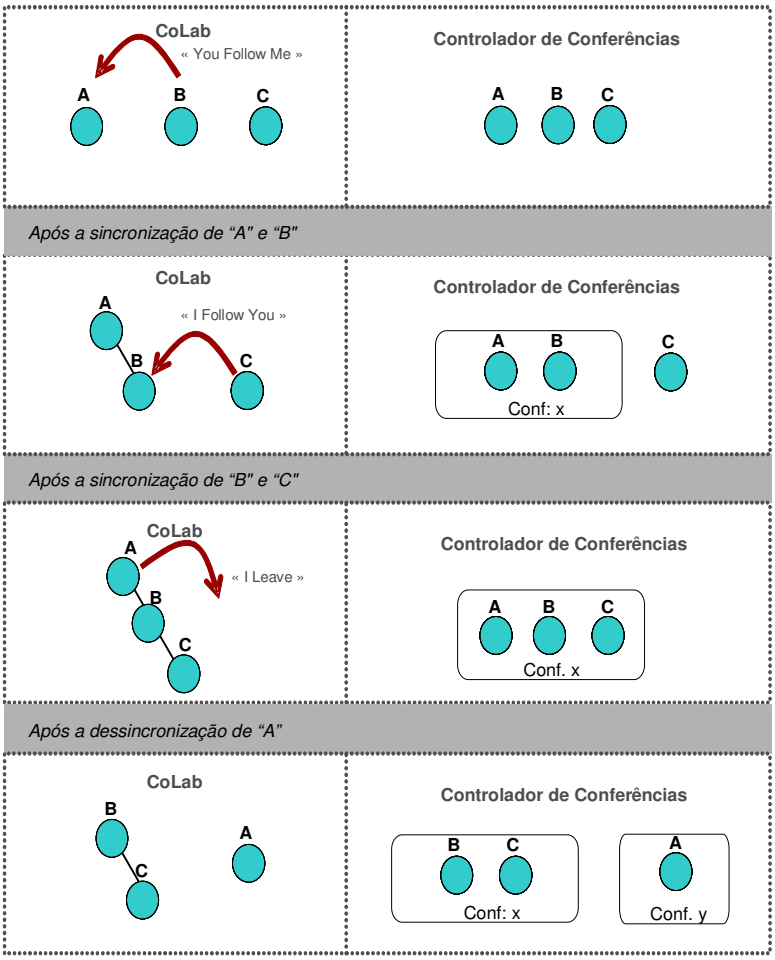

Figura 22 - Teste de Funcionalidade

6.6 Teste de Desempenho

Além do teste de funcionalidade, foi realizado um teste de desempenho do sistema para avaliar o tempo de resposta do Controlador de Conferências. O cenário foi feito para obtenção do tempo de resposta na criação de conferência com dois membros que se sincronizam em um grupo de trabalho CoLab. Os resultados dos testes são ilustrados na figura 23, onde mostram os tempos médios de 3 ensaios. Como visto na figura, o Controlador de Conferências envia um convite SIP para o primeiro usuário (carqueja), com tempo de 2,9s após o estabelecimento de uma sincronização CoLab entre dois usuários. Após 0,36s do recebimento do aceite do convite SIP, o Controlador de Conferências encaminhou o convite SIP para o segundo participante (cidreira).

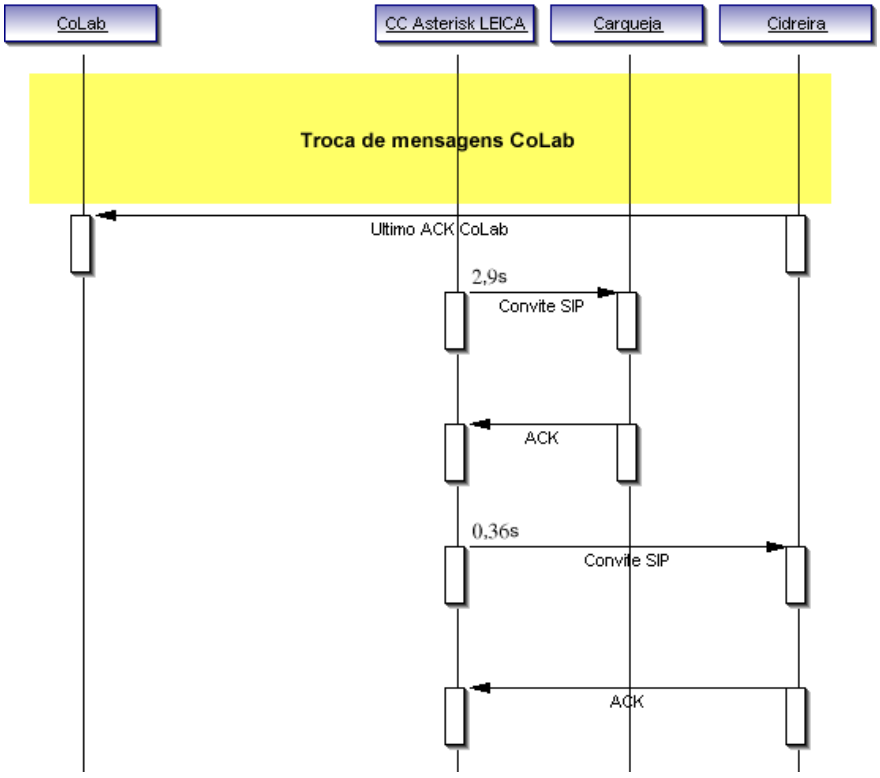

Figura 23 - Cenário de operação do sistema de co-navegação com suporte à conferência

Analisando os resultados, pode-se perceber que a ocorrência da primeira chamada SIP envolve um tempo relativamente alto (2,9s), mas que em termos de sincronização é

considerado aceitável. A segunda chamada ocorre logo após a primeira ter sido aceita e como o controlador se encontra na mesma máquina do servidor Asterisk, isto é feito rapidamente, levando apenas 0,36s.

6.7 Considerações Finais

Neste capítulo foi vista a integração do Controlador a outras ferramentas colaborativas. Os testes efetuados foram realizados no contexto da biblioteca digital do projeto SIDIE (CNPq-SIDIE, 2007). O objetivo foi testar sessões de ensino e busca colaborativa. Com as sessões de ensino, permitir ao professor realizar sessões de leitura com os alunos, enquanto na busca colaborativa permitir aos alunos investigarem o conteúdo da biblioteca, permitindo assim busca em conjunto. Na análise dos testes o Controlador se mostrou apto a interagir com o CoLab e apresentando a interação das sincronizações de co-navegação com as chamadas SIP. As funcionalidades previstas pelas políticas de colaboração foram adequadas ao ambiente proposto com o co-navegador, tendo em vista apenas uma validação da API do Controlador. Também foi visto o desempenho do controlador no ambiente, mostrando que mesmo a integração de sincronização com as chamadas SIP não ocorre de forma simultânea, compensa o seu uso por este tempo não interferir de forma substancial no trabalho dos participantes no ambiente. Como teste futuro poderia ser incluindo uma análise do atraso de LEICA na execução das regras e também fazer um estudo do porque o tempo para chamada do segundo participante ser tão inferior ao do primeiro.

7 CONCLUSÕES

O desenvolvimento de ambientes colaborativos tem se tornado uma área de pesquisa importante, uma vez que permite que um grupo de pessoas trabalhe em tarefas ou objetivos em comum, independentemente de sua localização geográfica. Diferentes tipos de ferramentas podem ser incluídas em um ambiente colaborativo, cada uma delas com diferentes características e níveis de interação entre os usuários.

Na criação de ambientes colaborativos, entretanto, as necessidades dos usuários podem variar de acordo com suas características e/ou as tarefas que os mesmos devam realizar, e, portanto, não há uma solução única que possa atender todos os possíveis requisitos. Neste sentido, tornam-se interessantes ambientes extensíveis, que venham a permitir que ferramentas específicas sejam a eles acopladas, de maneira a atender as necessidades particulares de cada projeto.

Uma das limitações de sistemas colaborativos atuais são as soluções de comunicação especificamente desenvolvidas para serem integradas a estes sistemas. Além disso, torna-se inviável a substituição destas ferramentas de comunicação oferecidas por outras disponíveis no mercado, comprometendo a flexibilidade desses sistemas.

No capitulo três foi apresentado LEICA, uma ferramenta que provê uma integração de funcionamento fracamente acoplada, permitindo a abstração dos detalhes internos de cada aplicação. Assim, o processo de integração é facilitado. Como forma de validação, LEICA permitiu a integração do Controlador de Conferências ao CoLab, fazendo com que o mesmo funcionasse de maneira dinâmica na criação e destruição de conferências.

A comunicação por voz é uma das opções que podem tornar os ambientes colaborativos mais atrativos. Uma vez que os usuários encontram-se, em geral, geograficamente separados, a utilização de meios convencionais de comunicação por voz torna-se inviável financeiramente. Neste sentido a comunicação por sistemas de voz sobre IP (VoIP) vem atraindo, atualmente, bastante interesse, uma vez que pode proporcionar comunicação por voz com boa qualidade e baixo custo. Assim, acredita-se que a integração de ferramentas de VoIP em ambientes colaborativos seja uma alternativa viável.

Neste trabalho foi proposto um Controlador de Conferências de voz, baseado nas tecnologias de VoIP, que pode ser integrado facilmente a outras aplicações colaborativas. Devido ao fato de se utilizar de tecnologias em evidência e em constante desenvolvimento, como é o caso de VoIP, as conferências de voz integradas em ambientes colaborativos devem se beneficiar das melhorias que venham a ocorrer em ambientes de telefonia IP.

O Controlador proposto foi concebido baseando-se na premissa de que fosse facilmente acoplável em ferramentas de integração de ambientes colaborativos, optando-se pela utilização do LEICA para esta finalidade. Adotou-se como servidor SIP o Asterisk, uma solução gratuita e de código-aberto, com larga utilização em telefonia sobre IP.

Para viabilizar o desenvolvimento de uma ferramenta que fizesse acesso ao servidor Asterisk, foi utilizado o pacote Asterisk-Java, que provê suporte à integração e controle do servidor Asterisk, sendo adotada, assim, a linguagem Java para a implementação. O Asterisk, por sua vez, foi configurado especialmente para o uso do controlador. Foram utilizadas ainda a arquitetura de tempo real (ARA), que permite o gerenciamento de usuários SIP, e a Manager API, responsável por controlar o Asterisk remotamente.

Uma das características que trazem vantagem na utilização do Controlador apresentado é sua utilização em conjunto com outras ferramentas, de forma a proporcionar uma atuação mais efetiva dos usuários neste tipo de ambiente. Além da navegação colaborativa e de ferramentas de chat, as conferências de voz trazem uma maior interatividade entre os envolvidos no desenvolvimento de uma tarefa.

Como forma de validação do Controlador de Conferências proposto, foi apresentado um estudo de caso onde foi feita a integração de CoLab, uma ferramenta de co-navegação, com o Controlador de Conferências, através da ferramenta de integração LEICA.

A integração apresentada permitiu que operações de sincronização dos membros das sessões CoLab passassem a controlar a entrada e saída destes membros em diferentes conferências de voz. A automatização desta operação evita que os usuários tenham que controlar manualmente qual conferência devem chamar para discutir o assunto co-navegado, bem como identificar o momento em que devem deixar a conferência, uma vez que a sessão CoLab tenha sido finalizada. Estas ações facilitam, portanto, a gerência das sessões.

A validação do Controlador incluiu testes de funcionalidade e testes de desempenho entre estas duas ferramentas. Os testes de funcionalidade apresentaram três usuários utilizando CoLab junto ao um softfone para permitir que navegassem em conjunto sempre na mesma conferência caso estivessem sincronizados. Os testes de desempenho apresentaram os tempos de espera decorridos entre a etapa de sincronização da co-navegação e a chamada para ingresso na conferência.

Além das contribuições apresentadas, acredita-se que esta dissertação abra possibilidades de trabalhos futuros que venham a complementar o funcionamento da ferramenta desenvolvida, bem como a prover possíveis melhorias no controlador proposto. Dentre as possibilidades de continuação do presente trabalho, pode-se citar aprimoramentos no mecanismo de controle de palavra no Controlador. Uma vez que estes mecanismos já estão implementados, uma semântica de integração mais elaborada poderá ser definida para a Supersessão, criando-se novas regras na sua Política de Colaboração, com o objetivo de associar privilégios ou permissões especiais a tipos diferentes de usuários (LEICA suporta a associação de papéis aos usuários de uma Supersessão). Um exemplo de melhoria decorrente desta ação seria permitir que apenas usuários do tipo “Professor” possam controlar a palavra em uma áudio-conferência.

É importante, também, verificar 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 servidores Asterisk. O problema da escalabilidade também deverá ser investigado no caso de CoLab, empregando-se por exemplo diferentes servidores Proxies para a co-navegação.

8 REFERÊNCIAS

APP_CONFERENCE. **Asterisk app_conference**. Disponível em: http://www.voip-info.org/wiki-Asterisk+app_conference acessado em Novembro de 2007.

ASTERISK. **The Open Source PBX IP.** Disponível em:http: //www.asterisk.org em Novembro de 2007.

ASTERISK-JAVA. **Java control for Asterisk PBXes.** Disponível em:http://asterisk-java.org acessado em Novembro de 2007.

BABYLON CHAT. **Babylon Java Chat.** Disponível em: http://visopsys.org/andy/babylon/index.html acessado em Novembro 2007.

BERNARD, S. **Spécification d'un environnement d'ingénierie collaborative mulitisite - Application à l'industrie aéronautique européenne**. Thèse de doctorat Génie Industriel École nationale supérieure d'arts et métiers, Centre d'Aix-en-Provence France, Nov.2004

CABRI, G., LEONARDI, L. ZAMBONELLI, F. **A Proxy-based Framework to Support Synchronous Cooperation on the Web**. Software – Practice and Experience 29(14), John Wiley & Sons, Ltd., 1999

CHONG, S.T., SAKAUCHI, M. **E-CoBrowse: co-Navigating the Web with Chat-pointers and Add-ins – Problems and Promises.** IASTED ICPDS, Collaborative Technologies Symposium, ACM, Las Vegas, NV, USA, Nov. 2000.

CNPq - SIDIE. **Sistema de Disponibilização de Informações para o Ensino.** Disponível em: http://www.literaturabrasileira.ufsc.br acessado em Novembro 2007.

ELLIS, C.A., GIBBS, S.J., REIN G. **Groupware: some issues and experiences.** Communications of the ACM , ACM Press v.34 , n1, pp.39-58, 1991.

GEROSA, L., GIORDANI, A., RONCHETTI, M., Soller, A., Stevens, R. **Symmetric Synchronous Collaborative Navigation.** IADIS International Conference WWW/Internet: p. 748-754, 2004.

GOTOMEETING. **Web Conference, Web collaboration, Collaboration software, Web Conferencing Software.** Disponível em: http://meetmenow.webex.com acessado em Novembro 2007.

GOMES, R.L., HOYOS-RIVERA, G.J., COURTIAT, J.P. **Um Ambiente para Integração de Aplicações Colaborativas.** Simpósio Brasileiro de Sistemas Colaborativos (SBSC 2006), Natal, RN. Nov, 2006.

GOMES, R.L., HOYOS-Rivera, G.J., Courtiat, J.P. **Integrating Collaborative Applications with LEICA.** 3rd IEEE International Conference on Information Technology: Research and Education (ITRE 2005), Taiwan, Jun, 2005.

GOMES, R.L., HOYOS-RIVERA, G.J., COURTIAT, J.P. **Loosely-coupled integration of CSCW systems.** in Proc. 5th IFIP Int. Conf. DAIS, Athens, Greece, , vol. 3543, pp. 38–49, Jun 2005

GONÇALVES, F. E. de A. **Asterisk PBX Guia de Configuração.** 1. Ed. V.Office São Paulo , 2005.

GRUDIN, Jonathan **CSCW: History and Focus**. IEEE Computer, IEEE Computer Society Press v.27, n.5, pp.19-26, 1994.

HAAKE, J.M., WIIL, U.K., NÜRNBERG, P.J. **Openness in shared hypermedia workspaces: The case for collaborative open hypermedia systems.** ACM SIGWEB Newsletter, ACM press v.8, n.3, pp.33-45, Oct. 1999.

HEARME. **Video and Voice Web Conferencing for Professionals.** Disponível em: http://www.hearme.com acessado em Novembro 2007.

HONG, H. C., CHEN, Y. C. **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.. **CoLab A New Paradigm and Tool for Browsing Collaboratively the Web.** IEEE Transactions on Systems, Man and Cybernetics. Part A, Systems and Humans, v.36, n.6, Nov, 2006.

HOYOS-RIVERA, G.J.; LIMA-GOMES, R., COURTIAT, J.P. **CoLab: Co-navigation sur le Web.** in Proc. NOuvelles TEchnologies de la REpartition (NOTERE), Toulouse, França. Jun, 2006.

HOYOS-RIVERA, G. J. **CoLab: Conception et mise en oeuvre d'un outil pour la navigation WEB**. Tese de Doutorado, Université Paul Sabatier. Toulouse, França, 2005.

IETF. ROSENBERG J, SCHULZRINNE H, CAMARILLO G,. JOHNSTON A, PETERSON J, SPARKS R, HANDLEY M,. SCHOOLER E. **SIP: Session Initiation. Protocol**. RFC 3261, Internet Eng. Task Force, 2002.

IETF. GREENE, N.; RAMALH, M; ROSEN, B., **MGCP: Media Gateway Control Protocol Architecture and Requirements**. RFC 2805, Internet Eng. Task Force, 2000.

ITU_T. International Telecommunication Union Telecommunication Standardization Sector ITU-T, **Packet Based Multimedia Communications Systems.** Set, 1999.

IVISIT. **Real time multi-party vídeo conferencing and colaboration tools.** Disponível em: http://www.ivisit.com acessado em Novembro 2007.

JOHANSEN, R. **Groupware: Computer Support for Business Teams**. The Free Press 205p, 1988. ISBN 0-02-916491-5

JOHNSTON, A. **SIP-Understanding the session initiation protocol**. Ed. Artech House, 2ª Edição, 2001.

KOSKELAINEN P., SCHULZRINNE H., Wu X. **A SIP-based conference control framework.** NOSSDAV, p. 53-61, 2002.

KYNG, M.: **Design for Cooperation: Cooperation in Design**; Communications of the ACM; vol 7 nr 4, Dez, 1991.

KIM, D.H. PARK, S.M., KIM, J.Y., SUL, D.M., LEE, K.H. **Collaborative Multimedia Middleware Architecture and Advanced Internet Call Center.** ICOIN, pp. 246-252, 2001.

LAURILLAU, Y. **Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe : le modèle et la plate-forme Clover**. Thèse de doctorat Informatique Université Joseph Fourier - Grenoble I, Grenoble, França, Set. 2002

LIMA-GOMES, R. LEICA: **Un environnement faiblement couplé pour l'intégration d'applications collaboratives**. Tese de Doutorado, Université Paul Sabatier. Toulouse, França, 2006.

LIEBERMAN, H.; VAN DYKE, N.W., VIVACQUA, A.S. **Let's Browse: A Collaborative Web Browsing Agent.** IUI'99, ACM, Redondo Beach, CA, EUA, 1999.

MALY, K., ZUBAIR, M., Li, L. **CoBrowser: Surfing the Web Using a Standard Browser**. World Conference on Educational Multimedia, Hypermedia and Telecommunications, p. 1220-1225, 2001.

MAHLER, P: **VoIP Telephony with Asterisk**. Ed. Signate, 2ª Edição, 2004.

MANAGER API. **Asterisk Manager API.** Disponível em: http://www.voip-info.org/wiki-Asterisk+manager+API acessado em Novembro 2007.

MEGGELEN, J. V., SMITH, J., MADSEN, L.: **Asterisk: The Future of Telephony.** Ed. O´Reilly, 2005.

NIELSEN, J.; **Multimedia and Hypermedia – The Internet and Beyond**. Academic Press Inc., 1996.

NUNAMAKER, J.F., DENNIS, A.R., VALACICH, J.S., VOGEL, D.R., GEORGE, J.F. **Electronic meeting systems to support group work**. Communications of the ACM 34(7):40-61, 1991.

DOMMEL , P., GARCIA-LUNA-ACEVES, J.J. **Floor Control for Multimedia Conferencing and Collaboration.** ACM Multimedia Systems Journal, Vol. 5, No. 1, Jan, 1997.

ROSENBERG, J., SCHULZRINNE, H.: **Models for multiparty conferencing in SIP.** Internet Draft, Internet Engineering Task Force, IETF, 2000.

SIDLER, G.; SCOTT, A. WOLF, H. **Collaborative Browsing in the World Wide Web**. JENC8, pp. 221-1 – 221-8, 1998.

SINGH, K., SCHULZRINNE, H. **Unified Messaging using SIP and RTSP.** IP Telecom Services Workshop, Atlanta, Georgia, EUA, Set, 2000.

SIGGELKOW, N., Levinthal, D. **Temporarily divide to conquer: Centralized, decentralized.** Org. Sci., vol. 14, no. 6, p. 650–669, 2003.

SOARES, L.C., Freire, V.A.. **Redes Convergentes.** Ed. Altabooks, Rio de Janeiro. 2002.

SPENCER, M., ALLISON, M., RHODES, C. **The Asterisk Handbook version 2**. Digium. Disponível em http://www.digium.com acessado em Novembro 2007

SINGH, K, WU, X., LENNOX, J., SCHULZRINNE, H. **Comprehensive Multi-platform Collaboration.** MMCN 2004 - SPIE Conference on Multimedia Computing and Networking, 2004.

TANENBAUM, A. S. **Redes de Computadores**. 4ª Edição. Ed. Campus, São Paulo, 2003.

TERZIS, S., NIXON P. **Building the next generation groupware: A survey of groupware and its impact on the virtual enterprise.** Technical Report TCD-CS-1999-08, Department of Computer Science, Trinity College Dublin , Irlanda, 1999.

WEBEXMEETMENOW. **Web Meeting, Online Meeting, Presentations Online, Desktop Sharing.** Disponível em: http://meetmenow.webex.com acessado em Novembro 2007.

YEO, C.K., HUI, S.C., SOON, I.Y. **A Multimedia Call Centre on the Internet.** International Journal of the Computer, The Internet and Management, Vol. 9, No.1, pp. 1-12, 2001.