

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

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

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

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

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

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

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

### 1. Introdução

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

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

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

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

## 2. Trabalhos Relacionados

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

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

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

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

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

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

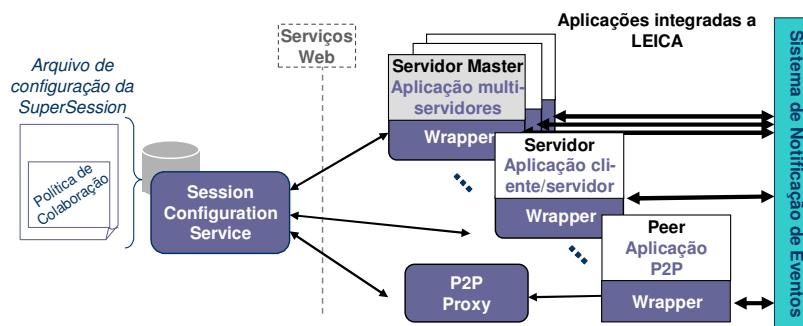
### 3. Ambiente de Integração LEICA

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

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

Com o objetivo de viabilizar a integração de diferentes aplicações colaborativas existentes, Gomes (2006) propôs LEICA – *Loosely-coupled Environment for Integrating Collaborative Applications*. Trata-se de um ambiente de integração que adota uma abordagem fracamente acoplada permitindo a abstração dos detalhes internos de cada aplicação, facilitando assim o processo de integração. Outra vantagem desta abordagem é que, uma vez integrada ao ambiente, a aplicação colaborativa mantém sua autonomia de execução. Da mesma forma, LEICA mantém-se independente das aplicações, isto é, novas aplicações podem ser integradas ou retiradas do ambiente sem comprometer a execução deste último. O grau de interação entre as aplicações integradas depende exclusivamente da API de cada aplicação, sendo que esta API caracteriza a natureza dos eventos que podem ser observados e as ações que cada aplicação é capaz de executar.

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



**Figura 1. Princípio arquitetural de LEICA**

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

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

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

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

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

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

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

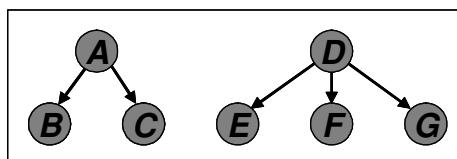


Figura 2. Cenários de configuração SDTs

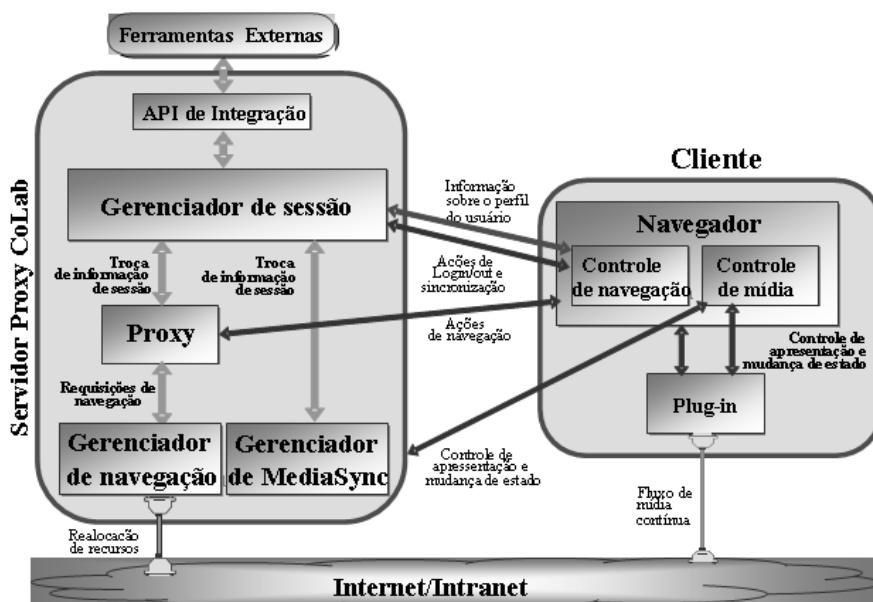
A Figura 3 apresenta a arquitetura CoLab que segue uma abordagem baseada na utilização de um servidor Proxy. Quando um usuário assíncrono solicita uma ação de navegação, esta requisição é enviada ao servidor Proxy CoLab. O servidor satisfaz a requisição e informa a ocorrência desta ação aos navegadores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

O servidor CoLab possui cinco módulos:

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

O cliente CoLab é composto de dois módulos:

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



**Figura 3. Arquitetura do CoLab**

## 5. Controlador de Conferências para Ambientes Colaborativos

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

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

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

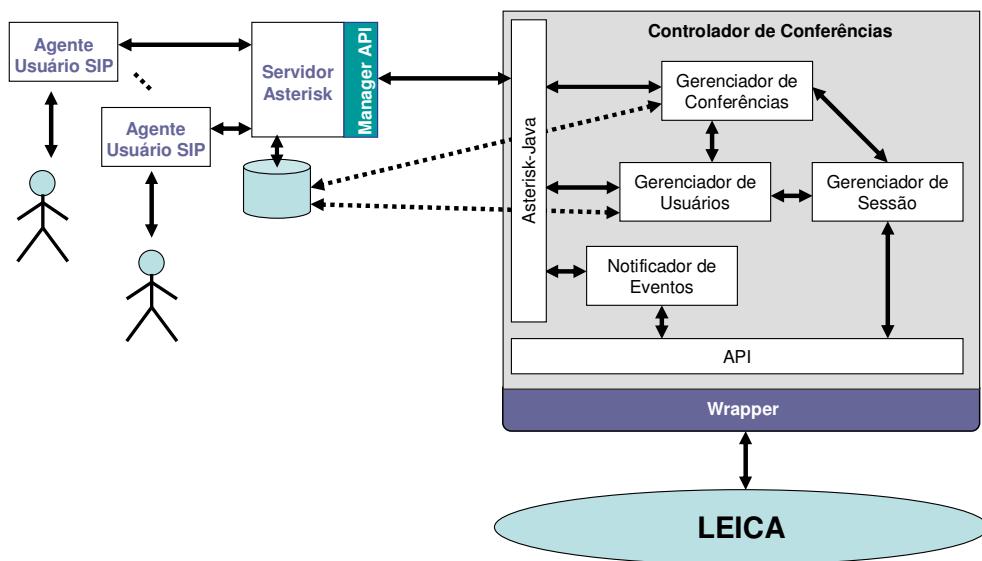
Outra característica do nosso Controlador de Conferência é o suporte apenas ao modelo de conferência *dial-out*, onde todo o gerenciamento de usuários e de conferências é diretamente realizado pelo Controlador em resposta a requisições recebidas das aplicações externas. Este controle automático simplifica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

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

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

### 5.1. Arquitetura do Controlador de Conferências

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



**Figura 4. Arquitetura do Controlador de Conferências**

As seções que seguem apresentam os principais componentes desta arquitetura.

### 5.2. Servidor Asterisk

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

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

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

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

### 5.3. Usuários e seus Agentes Usuários SIP

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

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

### 5.4. Controlador de Conferências

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

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

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

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

## 6. Co-navegação com Suporte a Conferência

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

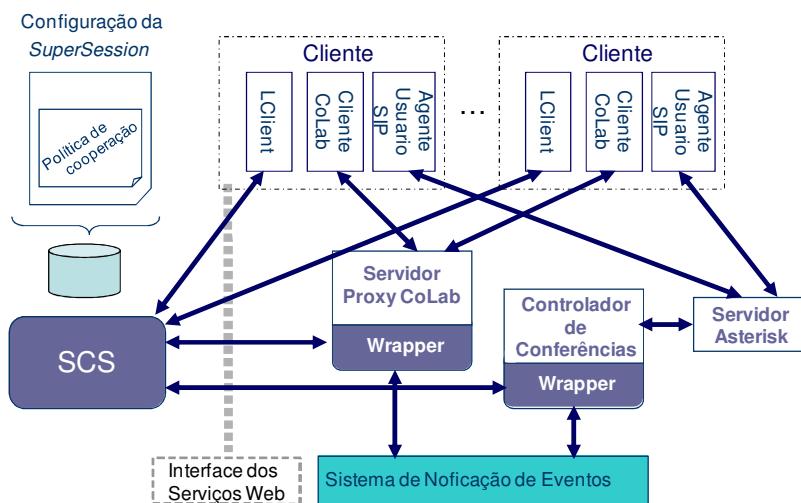
Para facilitar a definição de uma *SuperSession* global, é importante definir o conceito de sessão para cada sistema. Cada sessão CoLab deve, então, ser associada a uma sessão do Controlador de Conferências. No contexto de uma sessão CoLab, diferentes grupos podem ser formados durante a colaboração,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

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

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

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

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

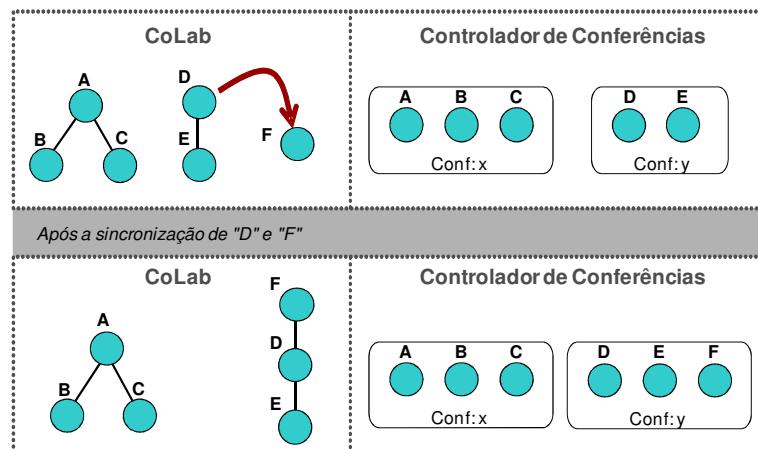


**Figura 5. Ambiente de Co-navegação**

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

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

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



**Figura 6. Cenário de Co-navegação**

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

## 7. Conclusões e Trabalhos Futuros

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

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

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

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

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

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

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

## Agradecimentos

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

## Referências

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

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