

3 Trabalhos Relacionados

Neste capítulo são apresentados os trabalhos relacionados ao CAS que, de algum modo serviram como inspiração e fonte de informações para o desenvolvimento desta pesquisa. Apesar de o tema Captura & Acesso ser recorrente na literatura, não foram encontrados trabalhos que abordam especificamente a proposta aqui apresentada de engenharia de documentos. Desta forma, nesta seção são focados alguns dos projetos mais relevantes e que foram considerados para uso como infra-estrutura básica para esta pesquisa.

3.1. eClass/Classroom 2000

O projeto Classroom 2000, também conhecido por eClass, foi desenvolvido no Georgia Tech e permite a captura de diferentes fluxos de informação, produzidos durante uma aula convencional realizada em uma sala de aula instrumentada com uma infra-estrutura de *hardware* e *software*. A captura não se restringe ao áudio e vídeo, mas inclui conteúdo gerado pelo professor como as apresentações Microsoft PowerPoint® e as anotações e URLs visitadas durante a aula. O sistema produz automaticamente documentos hipermídia que registram todo esse conteúdo capturado, com a apresentação por meio da web [AABE98].

O Classroom 2000 foi o precursor da maioria dos sistemas encontrados na literatura, mas as pesquisas do grupo, que vinham sendo realizadas desde 1995, mudaram de foco por volta de 2002 quando começaram a surgir sistemas comerciais como o Mimio [MIMI08] e o Silicon Chalk [Coat02, Gold02]. Atualmente o grupo se dedica a pesquisas de sistemas para o auxílio a crianças altistas², incluindo até sistemas de C&A nesta área específica como o CareLog

² <http://home.cc.gatech.edu/autism>

[KHWS07]. A continuidade das pesquisas originais coube a um grupo da Universidade de São Paulo de São Carlos (USP São Carlos), que colaborava com o eClass desde 1998, em um novo projeto chamado InCA-SERVE.

3.2. InCA-SERVE

Desenvolvido desde 2000, o projeto InCA-SERVE investiga os problemas associados à construção de aplicações de computação ubíqua que enfatizam captura e acesso a informações em ambientes tais como salas de aula e de reuniões [CARP03]. O projeto permite a captura e o acesso a informações multimídia evolucionárias mediante a construção de duas infra-estruturas complementares: InCA e SERVE.

A infra-estrutura InCA (*Infrastructure for Capture and Access*) foi construída para simplificar o desenvolvimento de aplicações de C&A de atividades genéricas, com uma estrutura escalável, extensível e flexível. Por ser a infra-estrutura de mais baixo nível, além de permitir a captura de atividades realizadas durante as sessões ao vivo, também provê um modelo de informações extensível para a captura e armazenamento de experiências interativas, dando suporte à informação multimídia evolucionária [TA04].

Ao explorar serviços da InCA, a infra-estrutura SERVE é responsável por armazenar (*Store*), estender (*Extend*), recuperar (*Retrieve*) e visualizar (*Visualize*) informações evolucionárias (*Evolutionary*). Essa infra-estrutura apresenta a informação capturada aos usuários e oferece serviços especializados fora da sessão de captura [Eleu02].

3.2.1. xInCA

O xInCA (*Extended Infrastructure for Capture and Access Applications*) [Cate04] é um projeto, também desenvolvido na USP São Carlos, que visa criar um ambiente genérico, estruturado e configurável para C&A de experiências do cotidiano que envolvam sessões. O xInCA é uma extensão da infra-estrutura InCA

baseada em componentes de *softwares* reutilizáveis. Ele é construído com componentes que implementam as funcionalidades mais comumente encontradas em aplicações de C&A (*whiteboard*, texto, áudio, vídeo e *web logging*) e é utilizado em dois projetos apresentados nas próximas seções: o eMeet e o iClass.

3.2.2. eMeet

O eMeet é um sistema distribuído de suporte a reuniões, que aplica vários conceitos do eClass de forma a contemplar o ambiente de uma reunião. O eMeet atende a grupos de usuários locais e distribuídos e é dividido em duas partes principais: o eMeet/InCA e o eMeet/SERVE, descritas a seguir:

O eMeet/InCA [Bila02] é responsável pela captura e armazenamento das informações geradas durante a reunião, ou seja: áudio e vídeo dos participantes, apresentações de *slides*, anotações e também a sumarização e sincronização intermídias capturadas. Além da captura, o *software* cliente é responsável pela difusão das mídias de áudio e vídeo de cada participante por RTP (*Real-Time Transfer Protocol*).

O eMeet/SERVE [Eleu02] é a porção responsável pelo suporte ao acesso, recuperação e apresentação dos dados capturados por meio do eMeet/InCA, gerando automaticamente hiperdocumentos multimídia a partir de informações capturadas em ambientes de reuniões distribuídos na web. Além disso, um mecanismo de sincronização foi implementado nos hiperdocumentos a fim de permitir uma reprodução adequada dos elementos multimídia. Estes hiperdocumentos são disponibilizados na web e podem ter seu acesso controlado.

A Figura 3.1 apresenta a arquitetura do sistema eMeet/SERVE. São várias camadas independentes, que permitem um sistema portátil em que a troca de um módulo como o SGDB, por exemplo, não gera impactos. Foi utilizada a linguagem PHP em seu desenvolvimento, mas dada a característica anterior de independência, outras linguagens podem ser usadas. No módulo de dados as mídias podem ser tanto armazenadas em um SGDB quanto no sistema de

arquivos. O módulo Aplicação eMeet/SERVE é o componente central e realiza tanto a geração do hiperdocumento, como provê o *web site* de acesso.

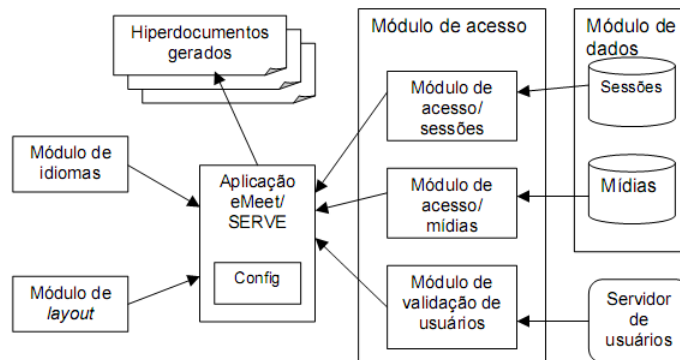


Figura 3.1: Arquitetura do sistema eMeet/SERVE [Eleu02].

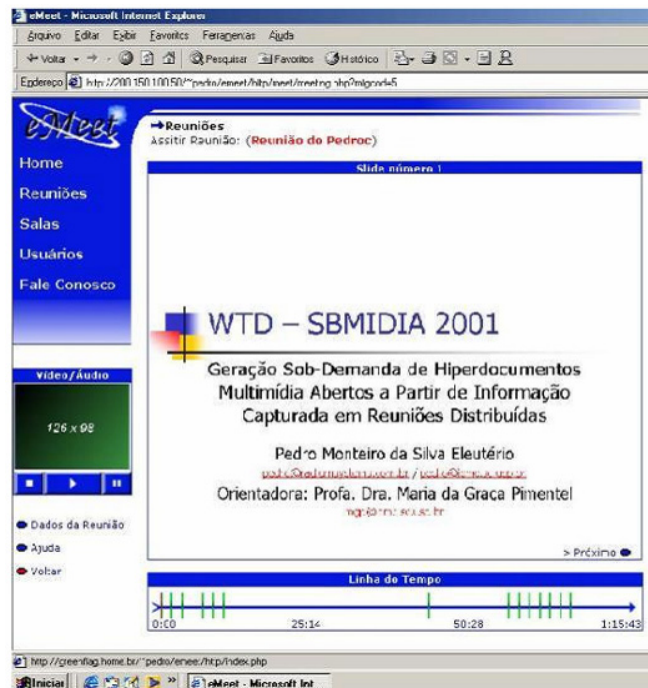


Figura 3.2: Visualização de uma sessão pelo eMeet/SERVE a partir dos dados capturados pelo eMeet/InCA [Eleu02].

Uma tela típica do eMeet/SERVE, apresentando um hiperdocumento gerado, é mostrada na Figura 3.2. Na parte central são apresentados os *slides* da reunião e uma linha do tempo com a marcação visual de suas transições. Na parte central à esquerda são apresentados o áudio e o vídeo capturados, de forma sincronizada com as anotações. A forma de sincronização escolhida, segundo

[Eule02], foi através de JavaScript com HTML e applet Java. O SMIL, opção comum em vários trabalhos correlatos, foi descartado pela baixa compatibilidade de *players*.

3.2.3. iClass

O iClass [CARP03] é uma evolução em termos de estruturação de *software* do sistema eClass e mantém o foco na captura de sessões em ambientes educacionais. Foi o projeto mais recente do grupo da USP São Carlos, dentro da arquitetura InCA-SERVE. Do ponto de vista de *software*, o projeto iClass é um conjunto de infra-estruturas e serviços escritos em linguagem Java e projetados com base nos sistemas InCA, xInCA, INDU [CARP03] e StRES [Bald02, CARP03] para simplificar a implementação de aplicações de C&A. Ele implementa funcionalidades do sistema eClass e adiciona outras como o suporte a dispositivos móveis.

O sistema de computação ubíqua iClass oferece um suporte efetivo à captura de experiências ao vivo. Para prover a captura de uma experiência, o sistema utiliza uma sala de aula equipada com microfones, câmeras de vídeo, projetores e um *smart board*. Cada aula é gravada com um único fluxo de áudio ou vídeo, e a interface de acesso disponibiliza diferentes formas de indexação das informações gravadas [CARP03].

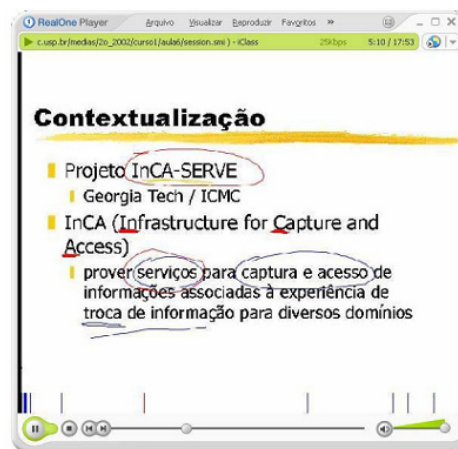


Figura 3.3: Apresentação SMIL da sessão capturada com o iClass no Real Player [CARP03].

O acesso às sessões capturadas pode ser realizado de formas distintas, visualizando um dos três tipos de documentos gerados automaticamente após a captura: SMIL, XHTML+SMIL e HTML padrão. A Figura 3.3 mostra uma sessão sendo apresentada em SMIL com *slides* e linha do tempo, em que cada marcador representa um *link* para o momento de visita a um *slide* [CARP03], sendo esta a única navegação possível.

3.3. ConferenceXP

Ao contrário dos demais projetos apresentados que eram de computação ubíqua ou especificamente de C&A, o projeto ConferenceXP³, uma iniciativa da Microsoft Research, é um *framework* extensível para o desenvolvimento de aplicações colaborativas e de ensino a distância. Ele permite que pesquisadores criem aplicações que se aproveitem dos recursos do ConferenceXP, assim como de tecnologias emergentes, como *displays* de alta resolução, Tablets PCs e redes sem fio, contempladas pela sua arquitetura. Mediante o uso de diversas APIs do *framework* base, o desenvolvimento de ferramentas colaborativas não precisa ser feito a partir do zero.

O ConferenceXP é focado para redes de alta velocidade com infra-estrutura de *multicast* como a Internet2⁴, para garantir eficiência e escalabilidade. O *framework* possui uma implementação customizada do protocolo de transporte em tempo real – RTP (*Real-Time Transfer Protocol*) – para transmissão com baixa latência [ABVV03]. O código do *framework* é totalmente gerenciado em cima do Microsoft .NET Framework[®] e integrado com as tecnologias de *media* e *display* DirectShow[®], DirectX[®] e Windows Media[®] também da Microsoft [BCHM04].

O ConferenceXP emprega uma arquitetura *peer-to-peer* (P2P) para não envolver servidores, tornando sua configuração fácil e prevenindo assim gargalos de tráfego e pontos únicos de falha. Sua arquitetura também é dividida em quatro

³ <http://www.conferencexp.net>

⁴ <http://www.internet2.edu>

camadas lógicas conforme ilustrado na Figura 3.4: Aplicação, Recursos, API e Transporte de Rede.

Segundo [BCHM04], as camadas de Aplicação e Recursos provêm a interface com o usuário para o ConferenceXP. Na camada de Recursos têm-se componentes que adicionam funcionalidades às aplicações. Por padrão o ConferenceXP já provê dois componentes: Áudio/Vídeo, que permite o envio e recebimento de fluxos de áudio e vídeo; e Presentation, que habilita o ConferenceXP a enviar e receber documentos e *inks*⁵. A camada de API é utilizada nestas camadas superiores para facilitar o desenvolvimento, já que encapsula diversos protocolos e tratamentos. O mesmo ocorre com a camada de transporte de rede, que garante que as transmissões das camadas superiores sejam feitas com o mínimo de perdas com sua implementação do RTP *peer-to-peer* e algoritmos de correção.

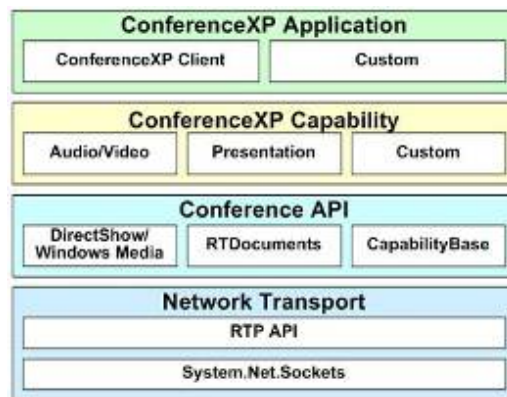


Figura 3.4: Arquitetura do ConferenceXP [BCHM04].

O ConferenceXP já inclui uma interface com o usuário, chamada de ConferenceXP Client, que permite a interação e colaboração com outras pessoas em um espaço colaborativo virtual (*venue*). Ao entrar num dos espaços disponibilizados deste cliente é iniciado automaticamente o recurso de áudio e vídeo multiponto em tempo real. Esta plataforma oferece, além de

⁵ *Ink* é a tinta virtual presente nos traços esboçados em uma lousa eletrônica ou em um *Tablet PC*. Trata-se de uma forma de entrada de dados nesses tipos de interface.

videoconferência grupo-a-grupo, um conjunto de ferramentas como troca de mensagens (*chat*), quadro branco eletrônico e apresentações distribuídas.

3.3.1. Classroom Presenter

Criado pela Universidade de Washington, o Classroom Presenter⁶ é uma aplicação desenvolvida com base no *framework* do ConferenceXP, que tem como objetivo viabilizar a educação a distância e o uso de salas de aula com recursos computacionais [SAW03]. Ele é um sistema de interação baseado em Tablets PCs que permite o compartilhamento de *ink* em *slides* entre instrutores e alunos, combinando as vantagens do estilo de aula com quadro-negro e apresentação de *slides*.

Com ele os professores podem projetar suas apresentações Microsoft PowerPoint® assim como transmiti-la em tempo real para os notebooks ou Tablets PCs dos alunos. Como ilustrado na Figura 3.5, os professores podem fazer anotações com *ink*, assim como destacar textos ou desenhar. Os alunos podem receber a apresentação do professor em tempo real em seus dispositivos, realizar suas próprias anotações e ainda fornecer respostas interativas [AACD06].

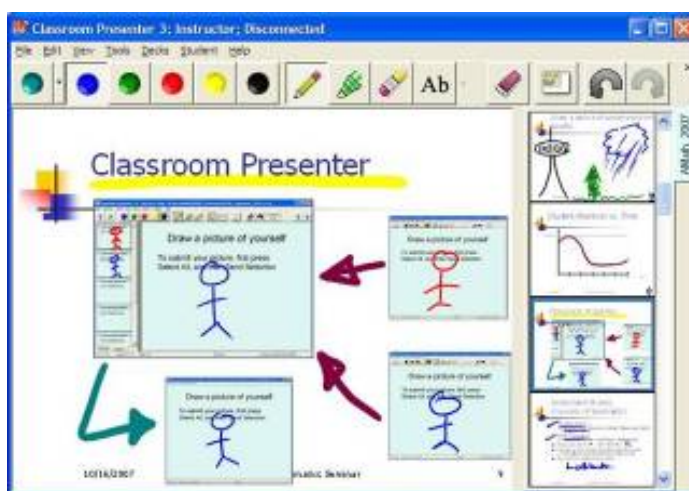


Figura 3.5: Visão do professor em uma aula com o Classroom Presenter [Clas08].

⁶ <http://classroompresenter.cs.washington.edu>

3.3.2. ConferenceXP Archive Service

O ConferenceXP Archive Service⁷ permite a gravação de uma sessão do ConferenceXP conectada a um servidor e o seu posterior acesso com a visualização em um dado espaço virtual (*venue*). É possível escolher quais fluxos RTP serão gravados, incluindo todos os componentes da camada de recursos como áudio e vídeo. O serviço também permite a exclusão de um fluxo já gravado. Todo o arquivamento da sessão é feito numa base de dados Microsoft SQL Server®.

3.3.3. ConferenceXP Web Viewer

O ConferenceXP Web Viewer⁸ é uma aplicação cliente, também desenvolvida pela Universidade de Washington, que permite a visualização de um fluxo de vídeo ao vivo ou gravado em sincronia com *slides* e anotações realizadas com *ink* em Tablet PCs. Ele embute o Microsoft Windows Media Player® e contém um subconjunto das funcionalidades do Classroom Presenter, suportando arquivos originários do ConferenceXP.

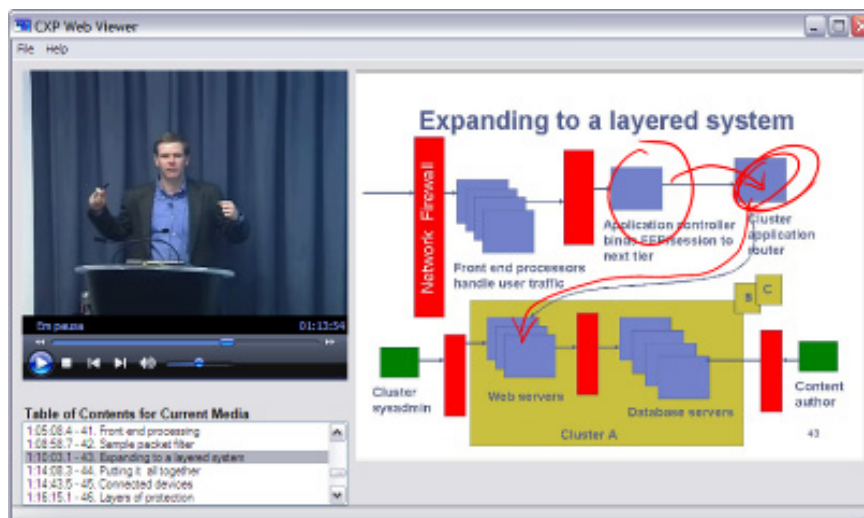


Figura 3.6: ConferenceXP Web Viewer.

⁷ http://research.microsoft.com/conferencexp/about_services_40.aspx

⁸ <http://www.cs.washington.edu/education/dl/confxp/webviewer.html>

Como observado na Figura 3.6, a visualização segue o paradigma de linha do tempo (*timeline*) e é apenas oferecida a possibilidade de navegação no evento baseada no título de cada *slide* e seus tempos relativos da apresentação. Todas as transições de *slides* são fielmente registradas e reproduzidas. Pode-se notar também na figura a reprodução das notas, com *ink*, na mesma ordem em que foram registradas.

O ConferenceXP Web Viewer se baseia num arquivo XML sem nenhuma característica de documento hipermídia para realizar a reprodução do evento. O programa é associado a arquivos com extensão WBV que contém apenas uma referência ao meta-arquivo ASX como ilustrado na Listagem 3.1. Este arquivo ASX, exemplificado na Listagem 3.2, possui a referência para o único vídeo que se pode associar ao evento e a um arquivo XML com o título dos *slides*, seus *timestamps* e as *inks* em formato binário. Como pode ser observado na Listagem 3.3, as mídias são referenciadas através de um *DeckGuid* que representa a pasta onde os *slides* estão salvos em formato JPEG com um número seqüencial.

```
<wmref href="Data\csep590tu_11_1.asx"> </wmref>
```

Listagem 3.1: Arquivo WBV associado ao ConferenceXP Web Viewer.

```
<ASX version = "3.0">
  <Entry>
    <Ref href = "csep590tu_11_1.wmv"/>
    <Title> CSEP 590TU, Lecture 11, Part 1 </Title>
    <PARAM name="CXP_SCRIPT" value="csep590tu_11_1.xml"/>
  </Entry>
</ASX>
```

Listagem 3.2: Arquivo ASX utilizado para referenciar o vídeo e o arquivo de scripts.

```
<?xml version="1.0"?>
<!-- CXP ArchiveTranscoder: 12/8/2005 11:52:14 AM -->
<WMBasicEdit >
<RemoveAllScripts />
<ScriptOffset Start="12/7/2005 18:35:35.58" />
<Options NoAutoTOC="false" PreferredWebViewerVersion="1.9.0.0" />
<Slides BaseURL="." Extent="jpg" />
<Title DeckGuid="107df139-29c4-4bd4-9d7c-7951b609913a" Index="0" Text="Security of
large systems University of Washington Homeland Security (UW CSE P590TU) 7
December 2005 Marty Lyons marty@cs.washington.edu"/>
```

```

...
<Title DeckGuid="107df139-29c4-4bd4-9d7c-7951b609913a" Index="42" Text="Expanding
to a layered system"/>
...
<Script Time="12/7/2005 19:38:00.59" Type="CXP3"
Command="AAEAAAD/////AQAAAAAAAAAMAgAAAEpBcmNoaXZlU1ROXYXsIFZlcnNpb249MS4wLjIwNTQuM
jQyNzksIEN1bHRlcmU9bmV1dHJhbCwgUHVibGljS2V5VG9rZW49bnVsbAUBAAAFUFyY2hpdmVSVE5hdi5
SVFVwZGF0ZQwAAAAHdmVyc2lrbG9yYWNrZ3JvdW5kQ29sb3IIZGVja0dlaWQHYmFzZVVyY2ZleHRlbnQ
WF0aW9uU2RlY2tBc3NyY2lhdG1vbG9iYWNrZ3JvdW5kQ29sb3IIZGVja0dlaWQHYmFzZVVyY2ZleHRlbnQ
Oc2NyY2xsUG9zaXRpb24Mc2NyY2xsRXh0ZW50AAAAAABBwEBAQAADggIBggCBgYCAAAAAQAAAAQAAA
AAAAAAPA/AAAAAYDAAAJDawMDAwMDAwLTawMDAtMDawMC0wMDAwLTawMDAwMDAwMDAwMDAwMAkEAAAABGU
AAAAKMTA3ZGYxMzktMj1jNC00YmQ0LT1kn2MtNzklMWI2MDk5MTNhBgYAAAAACQYAAAAAAAAAAAAAAAAA
AAAAAPg/DwQAAAC+AAAAAgABAAAA///wEAAAAAAAAADAIAAABU3lzdGvtLkRyYXdpbmcsIFZlcnNpb24
QEAAAUU3lzdGvtLkRyYXdpbmcuQ29sb3IEAAAABXZhbHV1Cmtub3duQ29sb3IFc3RhdGUEbmfzQAAAAE
JBwcCAAAAAAAAAAAAAACGAAEACgsL" />
...
</WMBasicEdit>

```

Listagem 3.3: Arquivo XML com os títulos dos *slides* e o script associado.

3.3.4. ConferenceXP Archive Transcoder

O ConferenceXP Windows Media Archive Transcoder⁹ é uma ferramenta desenhada para o pós-processamento dos dados de conferências e apresentações, preparando-os para *streaming* e uso *offline*. Ele é o responsável por pegar o conteúdo armazenado em bases SQL pelo ConferenceXP Archive Service, no formato do Classroom Presenter, e converter em um formato entendido pelo ConferenceXP Web Viewer.

3.3.5. ConferenceXP to Windows Media Gateway

O ConferenceXP to Windows Media Gateway¹⁰ é o responsável pela codificação e distribuição de um evento do ConferenceXP de forma que possa ser visto ao vivo em computadores pessoais com o Windows Media Player ou outro *player* de *streaming* compatível. Um típico usuário deste sistema é um aluno que não está em uma das salas com recursos de *multicast* conectadas com o ConferenceXP, mas ainda assim deseja assistir à aula remotamente por meio de sua conexão de banda larga.

⁹ http://www.cs.washington.edu/education/dl/confxp/archive_transcoder.html

¹⁰ <http://www.cs.washington.edu/education/dl/confxp/wmgateway.html>

Como o cliente apenas recebe um *streaming* de áudio e vídeo, não existe interatividade, não sendo, portanto, a forma ideal para aulas a distância. Apesar disso, devido à necessidade mínima de *software* (Internet Explorer e Windows Media Player disponíveis em qualquer sistema operacional Windows), é uma ferramenta muito útil para uso ocasional por participantes que não possam se deslocar até uma sala com todos os recursos exigidos pelo ConferenceXP.

O Windows Media Gateway implementa um tipo especial de nó do ConferenceXP que apenas recebe os fluxos *multicast* dos espaços colaborativos virtuais (*venues*) e demais nós. Um operador configura a aplicação para capturar os fluxos RTP de áudio e vídeo de interesse. A partir de então esses fluxos são transcodificados por meio do Windows Media Format SDK e distribuídos aos clientes que estão usando o Windows Media Server.

3.4. Considerações sobre os Sistemas

Neste capítulo foram apresentados os dois trabalhos mais atuais, completos e relevantes relacionados ao tema abordado: o InCA-SERVE e o ConferenceXP. Existem muitos outros sistemas de C&A, alguns já mencionados no Capítulo 2.3, sendo que todos têm particularidades e inovações interessantes. Apenas estes dois foram considerados para o uso neste trabalho, pela sua abrangência e devido à indisponibilidade dos demais, seja por descontinuidade dos projetos e/ou falta de documentação.

O projeto InCA-SERVE apesar de possuir muita documentação na forma de publicações, não é mais um projeto ativo, tendo a equipe migrado para outras áreas de pesquisa. Com a indisponibilidade tanto do *site* do projeto – o Coweb¹¹ – quanto do código fonte, o InCA-SERVE só serviu como inspiração para boas idéias quanto à estruturação da infra-estrutura e dos componentes para o CAS.

Já o ConferenceXP é muito mais bem documentado do que o InCA-SERVE e tem os códigos fontes e binários distribuídos em instaladores oferecidos no *site*

¹¹ <http://coweb.icmc.sc.usp.br/incaserve>

do projeto. Existe uma comunidade muito ativa tanto em fóruns de discussão quanto desenvolvendo aplicações para ele. Um aspecto negativo é a necessidade de uso de uma rede *multicast*, o que compromete a sua facilidade de configuração e uso. O ConferenceXP também não oferece suporte para câmeras IP utilizadas nos ambientes para o qual este projeto é focado e sua infra-estrutura teria que ser modificada para trabalhar com um documento hipermídia de captura com suporte a modelos navegacionais. Como seriam necessários muitos ajustes para o seu uso e com o complicador do *multicast*, optou-se por não utilizá-lo.

Sem nenhum projeto para servir de base da infra-estrutura, foi desenvolvido o CAS conforme detalhado na próxima seção, reaproveitando apenas algumas idéias desses projetos. Com o objetivo de estender o CAS a outros sistemas que serão usados na Active Classroom baseados no ConferenceXP, como o *Interactive Presenter for Handhelds* (iPH)¹² [Malc07], é possível desenvolver componentes que traduzam a comunicação entre os dois projetos. Quanto às propostas de engenharia de documentos, estas não foram encontradas em nenhum dos sistemas pesquisados, sendo uma inovação deste trabalho.

A parte da questão da infra-estrutura de base, comprovamos, com esse estudo mais profundo, que, no quesito de engenharia de documentos, todos os projetos de C&A analisados estão intrinsecamente ligados ao paradigma de linha do tempo (*timeline*). Dentro do projeto InCA-SERVE, as últimas iniciativas como o iClass já até utilizam uma linguagem hipermídia, no caso SMIL. No ConferenceXP, apesar de ser um trabalho mais recente, não há preocupação em se construir um documento hipermídia e utiliza-se um arquivo XML apenas com as marcações de tempo para mudança dos *slides*.

O NCL, linguagem utilizada nesse trabalho, é baseado no modelo conceitual hipermídia NCM (*Nested Context Model*), que, como reportado por [Soar00], estende a sintaxe e a semântica do SMIL, observado em vários sistemas citados nesta dissertação. O NCM é mais rico e segundo [Anto00] permite dentre outras coisas: a estruturação lógica do documento pelo uso de composições aninhadas; a

¹² <http://www.lac.inf.puc-rio.br/iph/>

definição de relacionamentos causais e de restrição entre seus componentes; a especificação do comportamento temporal e espacial do documento; e o uso de linguagens de *script* como Lua¹³ [Ieru06].

¹³ <http://www.lua.org>