

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**CAPTURA MULTIMÍDIA EM AMBIENTES EDUCACIONAIS
INSTRUMENTADOS: ASPECTOS ARQUITETURAIS,
MODELO DE COMUNICAÇÃO E INTERFACE DE ACESSO
CONTEXTUAL**

HIRAN NONATO MACEDO FERREIRA

Uberlândia - Minas Gerais

2012

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



HIRAN NONATO MACEDO FERREIRA

**CAPTURA MULTIMÍDIA EM AMBIENTES EDUCACIONAIS
INSTRUMENTADOS: ASPECTOS ARQUITETURAIS,
MODELO DE COMUNICAÇÃO E INTERFACE DE ACESSO
CONTEXTUAL**

Dissertação de Mestrado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Sistemas de Computação.

Orientador:

Prof. Dr. Renan Gonçalves Cattelan

Uberlândia, Minas Gerais

2012

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada “**Captura Multimídia em Ambientes Educacionais Instrumentados: Aspectos arquiteturais, modelo de comunicação e interface de acesso contextual**” por **Hiran Nonato Macedo Ferreira** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 25 de Junho de 2012

Orientador:

Prof. Dr. Renan Gonçalves Cattelan
Universidade Federal de Uberlândia

Banca Examinadora:

Prof. Dr. Renato de Freitas Bulcão Neto
Universidade Federal de Goiás

Prof. Dr. Pedro Frosi Rosa
Universidade Federal de Uberlândia

Profa. Dra. Sandra Aparecida de Amo
Universidade Federal de Uberlândia

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Data: 25 de Junho de 2012

Autor: **Hiran Nonato Macedo Ferreira**
Título: **Captura Multimídia em Ambientes Educacionais Instrumenta-
dos: Aspectos arquiteturais, modelo de comunicação e interface
de acesso contextual**
Faculdade: **Faculdade de Computação**
Grau: **Mestrado**

Fica garantido à Universidade Federal de Uberlândia o direito de circulação e impressão de cópias deste documento para propósitos exclusivamente acadêmicos, desde que o autor seja devidamente informado.

Autor

O AUTOR RESERVA PARA SI QUALQUER OUTRO DIREITO DE PUBLICAÇÃO DESTE DOCUMENTO, NÃO PODENDO O MESMO SER IMPRESSO OU REPRODUZIDO, SEJA NA TOTALIDADE OU EM PARTES, SEM A PERMISSÃO ESCRITA DO AUTOR.

Dedicatória

Aos meus pais, pelo apoio constante.

Agradecimentos

A Deus, por me capacitar e pela sua presença incondicional em minha vida.

Ao meu orientador Renan Gonçalves Cattelan, pela força, apoio e incentivo ao longo deste projeto.

À minha família, em especial meus pais, por todo amor que me foi dado e por estarem sempre do meu lado.

Aos professores Pedro Frosi Rosa e Sandra Aparecida de Amo, pelas sugestões e colaborações realizadas no decorrer deste trabalho.

Aos colegas da Faculdade de Computação, pelo companheirismo.

Aos alunos da Iniciação Científica, pelo apoio técnico no desenvolvimento e avaliação das arquiteturas.

À minha namorada Mônica, pelo carinho e paciência durante todo esse tempo.

Aos amigos de Uberlândia, em especial aos amigos do Ministério Universidades Renovadas, que foram como uma família para mim.

Ao CNPq, pelo apoio financeiro (146964/2010-1).

Enfim, a todos que contribuíram direta ou indiretamente para a realização deste trabalho.

*“Tu és, Senhor, o meu pastor,
por isso nada em minha vida
faltar .” (Salmo 23, 1)*

Resumo

Aplicações de captura e acesso dão suporte à captura automática de experiências do cotidiano e na posterior disponibilização das informações capturadas. Essas aplicações exploram o conceito de computação ubíqua para automatizar o processo de captura de atividades humanas do cotidiano e permitir a geração de documentos multimídia passíveis de armazenamento, recuperação, visualização e extensão ao longo do tempo. Seu desenvolvimento contempla a implantação de arquiteturas computacionais extensas e complexas, aspectos de modelagem e prototipação, bem como questões sobre armazenamento, recuperação e extensão das informações capturadas. Este trabalho apresenta o *Classroom eXperience*, uma infraestrutura de captura e acesso voltada para ambientes educacionais instrumentados que permite a gravação de uma série de artefatos produzidos durante uma aula e na sua posterior disponibilização aos alunos. A infraestrutura desenvolvida compreende recursos de tratamento contextual para acesso ao conteúdo armazenado, em que recomendações de conteúdo podem ser feitas a partir do contexto de acesso de cada usuário. A infraestrutura desenvolvida contempla ainda um modelo de comunicação *peer-to-peer* que suporta o armazenamento, a transferência e a replicação do conteúdo capturado. Por meio da criação de protótipos funcionais, foram também conduzidos estudos de caso e análises qualitativas sobre o uso da infraestrutura.

Palavras chave: computação ubíqua, captura e acesso, conteúdo multimídia , redes *peer-to-peer*, ciência de contexto.

Abstract

Capture and access applications support automatic capture of everyday experiences and the subsequent availability of such captured information. These applications explore the concept of ubiquitous computing in order to automate the process of capturing human activities and allow the generation of multimedia documents that can be stored, retrieved, viewed and extended over time. Their development involves the deployment of large and complex computing architectures, aspects of modeling and prototyping, as well as issues regarding storage, retrieval and extension of the captured information. This work presents *Classroom eXperience*, an infrastructure for capture and access in instrumented educational environments which allows the recording of a series of artifacts produced during a conventional lecture and its subsequent availability to students. The developed infrastructure includes mechanisms for contextual access of captured information, where content recommendations can be made according to the access context of each user. The infrastructure still comprises a peer-to-peer communication model supporting the storage, transfer and replication of captured content. By means of functional prototypes, case studies and qualitative analysis were conducted to evaluate the use of the system.

Keywords: ubiquitous computing, capture and access, multimedia content, peer-to-peer networks, context-aware.

Sumário

Lista de Figuras	xix
Lista de Abreviaturas e Siglas	xxi
1 Introdução	23
1.1 Contextualização	23
1.2 Motivação	24
1.3 Objetivos	25
1.4 Metodologia	25
1.5 Estrutura da Dissertação	26
2 Computação Ubíqua e Captura e Acesso de Conteúdo Multimídia	27
2.1 Considerações Iniciais	27
2.2 Computação Ubíqua	28
2.2.1 Classificação das Aplicações de Computação Ubíqua	28
2.3 Captura e Acesso	30
2.4 Questões de Projeto de Aplicações de Captura e Acesso	31
2.4.1 Dimensões dos Projetos de Captura e Acesso	31
2.4.2 Domínio e Heterogeneidade	31
2.4.3 Privacidade das Informações	32
2.4.4 Captura e Acesso Embarcada	32
2.5 Ambientes de Captura e Acesso	32
2.5.1 Captura e Acesso em Salas de aula	33
2.5.2 Captura e Acesso em Salas de Reuniões	34
2.5.3 Captura e Acesso para Propósitos Gerais	34
2.6 Características Comuns às Aplicações de Captura e Acesso	35
2.7 Considerações Finais	36
3 Classroom eXperience	37
3.1 Considerações Iniciais	37
3.2 Arquitetura do <i>Classroom eXperience</i>	37
3.2.1 Pré-produção	39

3.2.2	Gravação ao vivo	40
3.2.3	Pós-produção	41
3.2.4	Acesso	43
3.3	Mecanismo de Tratamento Contextual	44
3.3.1	Módulo de Tratamento de Consultas	45
3.3.2	Módulo de Tratamento de Restrições	46
3.3.3	Módulo de Apresentação	47
3.4	Visão Geral da Interação Entre as Etapas do Sistema	48
3.5	Considerações Finais	51
4	Modelo de Comunicação Utilizado no <i>Classroom eXperience</i>	53
4.1	Considerações Iniciais	53
4.2	Arquitetura	54
4.3	Especificação do Protocolo	55
4.3.1	Ambiente	56
4.3.2	Vocabulário	56
4.3.3	Formato	58
4.3.4	Serviços e Comportamentos	59
4.4	Detalhes de Implementação	64
4.5	Considerações Finais	66
5	Avaliação de Usabilidade	69
5.1	Considerações Iniciais	69
5.2	Avaliação de Usabilidade	69
5.2.1	Avaliação Heurística	71
5.2.2	Avaliação por Questionários	72
5.3	Avaliação de Usabilidade do <i>Classroom eXperience</i>	74
5.3.1	Resultado da Avaliação Heurística	74
5.3.2	Resultado da Aplicação de Questionários	75
5.4	Considerações Finais	77
6	Trabalhos Relacionados	79
6.1	Considerações Iniciais	79
6.2	Captura e Acesso de Experiências	79
6.3	Modelos de Comunicação para Aplicações de Captura e Acesso	83
6.4	Personalização e Recomendação de Conteúdo	85
6.5	Considerações Finais	87
7	Conclusão	89
7.1	Visão Geral, Resultados e Contribuições	89

7.2	Limitações	90
7.3	Trabalhos Futuros	90
Referências Bibliográficas		93
A Diagrama MER do CX		101
B Gráficos dos Resultados dos Questionários SUMI		103
A Questionário SUMI		105

Lista de Figuras

2.1	Dispositivos de computação ubíqua (lousa eletrônica, <i>Tablet PCs</i> , caneta eletrônica).	29
3.1	As quatro fases no processo de captura e acesso.	39
3.2	Aula sendo ministrada com recursos do <i>Classroom eXperience</i>	41
3.3	Documento XML gerado pelo componente de sincronização.	42
3.4	Trecho de um documento NCL gerado pelo componente de transformação.	43
3.5	Dimensões de contexto tratado no <i>Classroom eXperience</i>	45
3.6	Mecanismo de Tratamento Contextual.	45
3.7	Recomendação baseada nas preferências do usuário.	47
3.8	Diferentes formas de apresentação.	48
3.9	Interação entre as etapas da aplicação.	49
3.10	Tela de cadastro de uma aula (Pré-produção).	50
3.11	Área do aluno para acesso às aulas capturadas.	51
4.1	<i>Advertisement</i> para representação de um <i>pipe</i> na rede.	55
4.2	Ambiente com <i>peers</i> produtores e consumidores.	57
4.3	Primitiva para o protocolo proposto.	58
4.4	Autômato do serviço <i>start_session</i> [Araújo et al. 2012].	59
4.5	Autômato do serviço <i>end_session</i> [Araújo et al. 2012].	60
4.6	Autômato do serviço <i>list_status</i> (adaptada de [Araújo et al. 2012]).	61
4.7	Diagrama temporal para o serviço <i>list_status</i> (adaptada de [Araújo et al. 2012]).	62
4.8	Autômato para o serviço <i>ft_get</i> (adaptada de [Araújo et al. 2012]).	62
4.9	Autômato para o serviço <i>ft_put</i> (adaptada de [Araújo et al. 2012]).	63
4.10	Disposição das camadas.	65
6.1	Professor e aluno interagindo entre si através do sistema <i>Terakoya</i> [Nishiuchi et al. 2010].	80
6.2	Protótipo do sistema <i>Tele-Board</i> [Gumienny et al. 2011].	82
6.3	Disposição da infraestrutura de rede do sistema proposto por [Giliberti et al. 2011].	85

Lista de Abreviaturas e Siglas

API: Application Programming Interface
AVA: Ambiente Virtual de Aprendizado
AVC: Acidente Vascular Cerebral
C&A: Captura e Acesso
C/S: Cliente/Servidor
CAL: Content Abstraction Layer
CPref-SQL: Conditional Preference SQL
CX: Classroom eXperience
HTML: Hypertext Markup Language
IHC: Interação Humano-Computador
IP: Internet Protocol
JAXB: Java Architecture for XML Binding
JWS: Java Web Start
LOM: Learning Object Metadata
MD5: Message-Digest Algorithm 5
NAT: Network Address Translation
NCL: Nested Context Language
P2P: Peer-to-Peer
PC: Personal Computer
RFID: Radio Frequency IDentification
SBTVD: Sistema Brasileiro de Televisão Digital
SCORM: Sharable Content Object Reference Model
SQL: Structured Query Language
SUMI: Software Usability Measurement Inventory
TV: Televisão
VNC: Virtual Network Computing
XML: Extensible Markup Language
XMPP: Extensible Messaging and Presence Protocol
XSLT: eXtensible Stylesheet Language for Transformation

Capítulo 1

Introdução

1.1 Contextualização

Computação ubíqua pode ser considerada a maneira de disponibilizar recursos computacionais integrados ao ambiente de forma transparente, com o intuito de auxiliar os usuários na realização de suas tarefas diárias [Weiser 1991]. Esses recursos computacionais devem ser implantados no ambiente de modo não intrusivo, ou seja, os usuários que fazem o uso desses recursos não precisam alterar a forma como realizam suas atividades. A computação ubíqua visa a integração de dispositivos com diferentes funcionalidades e propósitos, a fim de proporcionar uma maior interação entre as pessoas que utilizam essas tecnologias.

Captura e acesso (C&A) é um tema de pesquisa recorrente na computação ubíqua, que tem como foco utilizar ferramentas especializadas para capturar informações do cotidiano das pessoas e prover recursos para os usuários acessarem esses conteúdos posteriormente. Muitas vezes, pessoas precisam realizar anotações de situações vividas para consultas posteriores; isso pode ser, não raro, uma tarefa bastante difícil. A captura pode acontecer de forma automática, gerando, ao final de cada sessão, documentos que possam ser acessados pelos usuários. Todo o conteúdo gerado precisa ser armazenado, recuperado, visualizado e estendido ao longo do tempo. Assim, as aplicações de C&A surgem para auxiliar pessoas nessa tarefa de registrar informações.

São diversos os domínios em que podem ser aplicados os conceitos de C&A. Algumas pesquisas focam no desenvolvimento de sistemas que têm a característica de capturar informações em reuniões [Chiu et al. 2001, He et al. 2003], outras, em atividades de colaboração [Kunz et al. 2010, Gumienny et al. 2011], outras, ainda, em captura de experiências pessoais [Wu et al. 2007], entre vários outros domínios.

Ambientes educacionais são um dos cenários propícios para a aplicação de técnicas de C&A. Diversos trabalhos propõem ferramentas para apoiarem professores e alunos na realização de suas atividades [Dickson et al. 2006, He e Zhang 2007, Zhang et al. 2008,

Nishiuchi et al. 2010, Zeng et al. 2010]. Nesses ambientes, geralmente são empregados mecanismos para capturar diferentes fluxos de mídias — áudio, vídeo e anotações feitas sobre lousas eletrônicas — e disponibilizar essas informações posteriormente aos alunos, com o intuito de auxiliá-los no processo de aprendizagem.

1.2 Motivação

Ambientes que envolvem apresentação de informações para um conjunto de pessoas usualmente requerem que os participantes realizem anotações sobre o conteúdo que foi exposto para consultas ou estudos posteriores. Muitas vezes, essas anotações demandam tempo e, principalmente, desviam a atenção do foco principal, que é o conteúdo que está sendo apresentado. Com isso, os ambientes de C&A surgem como uma proposta para auxiliar usuários no processo de captura e disponibilização dessas informações.

Um ambiente de C&A compreende a implantação de uma arquitetura computacional extensa e complexa, sofisticados canais para os fluxos de dados e de controle, e interfaces externas que facilitem a interação do usuário com o ambiente [Modahl et al. 2004]. Essas arquiteturas são consideradas extensas pois devem abordar diferentes frentes de pesquisa para desenvolvimento de alguma solução e complexa por exigir integração de diferentes recursos computacionais para captura e disponibilização das informações. A construção dessas aplicações contempla ainda aspectos de modelagem e prototipação de aplicações e desenvolvimento de abstrações para comunicação entre componentes distribuídos.

A forma como o conteúdo capturado será transmitido pela rede merece uma certa atenção. Essas infraestruturas devem fornecer mecanismos eficientes para comunicação, que permitam que os dados sejam trafegados de forma simples, segura e transparente. Nessa perspectiva, aplicações de C&A devem prover formas para a comunicação acontecer o mais simples possível. Esses mecanismos são necessários para permitir que os dados capturados sejam distribuídos na rede, possibilitando assim, uma forma mais estruturada de organizar os dados.

Outro importante desafio na construção de aplicações de C&A é o modo como os dados serão acessados pelos usuários, pois essas aplicações devem fornecer interfaces simples, intuitivas e organizadas para apresentação do conteúdo. Além dessas características, esses ambientes devem apresentar suporte à contextualização de acesso à informação, ou seja, a forma como o usuário está realizando acesso ao sistema interfere no que será apresentado a ele.

Dentre os diversos sistemas propostos para explorar o paradigma de C&A, destaca-se o *iClass* [Pimentel et al. 2007], uma plataforma de captura *open-source* que grava uma variedade de artefatos produzidos durante uma aula em uma sala instrumentada, e, ao final da gravação, gera documentos multimídia para apresentação ao usuário. O *iClass* utiliza uma abordagem de comunicação *publish-subscriber*, onde uma entidade centraliza

(*broker*) é responsável pela distribuição de todo o conteúdo produzido pelos componentes de captura. Essa abordagem de comunicação pode ser considerada uma limitação imposta neste sistema, pois caso essa entidade pare de funcionar, toda a infraestrutura estará comprometida. Outra limitação presente no *iClass* está na apresentação do conteúdo capturado ao usuário, pois as interfaces não permitem personalização de conteúdo.

1.3 Objetivos

Nesse sentido, este trabalho tem como objetivo geral a investigação de uma infraestrutura de computação ubíqua para a captura automatizada de atividades que estenda a plataforma *iClass* quanto ao tratamento dos problemas anteriormente expostos. Mais especificamente, além do planejamento e da implantação das instalações necessárias ao ambiente instrumentado, os objetivos específicos deste trabalho incluem:

- O desenvolvimento e a adaptação de componentes de software responsáveis pela captura dos diferentes fluxos de mídias disponíveis no ambiente (áudio, vídeo e anotações sobre os *slides*);
- A elaboração de extensões funcionais, possibilitando que o conteúdo produzido seja integrado em diferentes formatos e permita o acesso via plataformas Web e do Sistema Brasileiro de Televisão Digital (SBTVD);
- A criação de um arcabouço de comunicação apropriado para apoiar atividades de armazenamento e sincronização do conteúdo resultante de uma sessão de captura;
- Criação de mecanismos que integrem informações contextuais no processo de acesso via plataforma Web;
- Integração da infraestrutura proposta com um sistema de Banco de Dados com suporte a preferências, em que o processo de busca e recuperação de informações seja personalizado de acordo com as preferências de cada usuário.

1.4 Metodologia

Esta pesquisa foi desenvolvida sobre uma abordagem exploratória. Segundo [Wazlawick 2009] essa abordagem é adequada a domínios que são necessários realizar estudos de caso e análises qualitativas. Paralelamente a essa abordagem, tem-se o enfoque experimental, que é o método de pesquisa adequado à computação ubíqua [Weiser 1993]. Foram construídos protótipos funcionais da infraestrutura necessária em quantidade suficiente para depurar a viabilidade dos sistemas e foram realizadas análises de usabilidade buscando identificar o quão simples e intuitiva são as interfaces e a interação com a infraestrutura desenvolvida.

Para a realização das análises de usabilidade utilizou-se duas técnicas distintas. A primeira técnica é a avaliação heurística, que foi executada por um conjunto de avaliadores, profissionais na área de usabilidade, onde verificaram se a interface está de acordo com um conjunto de heurísticas, que são regras gerais para orientar a avaliação das interfaces conforme a usabilidade. Questionários de avaliação foi a outra técnica utilizada, essa é uma técnica bem estabelecida para coletar opinião de usuários [Preece et al. 2005].

A utilização desta metodologia de pesquisa tem sido confirmada na prática e a experiência tem demonstrado que, quando o progresso em uma área da computação ubíqua é atingido, por meio da experimentação com protótipos de aplicações, deve-se começar a investigar as abstrações de projeto relevantes para a classe de aplicações em questão e então desenvolver e implementar uma solução arquitetural que contemple tais abstrações [Modahl et al. 2004, Dey 2001, Truong 2005, Cattelan 2009].

1.5 Estrutura da Dissertação

O restante do texto está organizado da seguinte maneira: no Capítulo 2 são abordados os conceitos sobre computação ubíqua e captura e acesso de conteúdo multimídia, com o intuito de contextualizar melhor o leitor sobre os temas abordados. No Capítulo 3 são mostrados a arquitetura e os detalhes de implementação da infraestrutura desenvolvida. O modelo de comunicação utilizado nessa infraestrutura é exposto no Capítulo 4. No Capítulo 5 é descrito o estudo de usabilidade realizado para a avaliação das interfaces desenvolvidas. No Capítulo 6 é feito um levantamento sobre alguns trabalhos relacionados. Por fim, no Capítulo 7 são apresentadas as conclusões e contribuições do trabalho, juntamente com suas limitações e trabalhos futuros.

Capítulo 2

Computação Ubíqua e Captura e Acesso de Conteúdo Multimídia

2.1 Considerações Iniciais

A cada dia, os avanços tecnológicos criam novas propostas para integrar recursos computacionais à vida das pessoas. Esses recursos computacionais devem fazer parte do dia a dia das pessoas, auxiliando-as na realização de suas tarefas de forma mais eficiente e com maior comodidade. Uma abordagem proposta em 1991 por Mark Weiser, define que recursos computacionais devem ser integrados aos ambientes de forma transparente, permitindo que os usuários interajam com estes sem alterar a forma como desempenham suas tarefas. Essa abordagem ficou conhecida como computação ubíqua [Weiser 1991].

A computação ubíqua tem a característica de produzir recursos computacionais que se adaptem ao ambiente para ajudar usuários na realização de suas tarefas. Dentro da computação ubíqua, uma área de pesquisa que tem crescido são as aplicações de captura e acesso (C&A). Essa classe de aplicações provê aos usuários capacidade de registrar informações de experiências ao vivo e disponibilizá-las, posteriormente, para acesso e revisão do conteúdo capturado. A partir de recursos providos pelas aplicações de C&A, os usuários podem concentrar sua atenção completamente na experiência, sem se preocupar em registrar a informação.

Neste capítulo é apresentada uma visão geral sobre os temas de pesquisa abordados neste trabalho. Conceitos de computação ubíqua são mostrados na Seção 2.2. Na Seção 2.3 é apresentada a área de pesquisa de C&A de conteúdo multimídia. Na Seção 2.5 são descritos os principais domínios de pesquisa em C&A. Na Seção 2.4 são abordadas as características dessas aplicações. Na Seção 2.6 são expostos os principais objetivos dos projetos em C&A. Por fim, na Seção 2.7 são feitas as considerações finais.

2.2 Computação Ubíqua

No surgimento dos primeiros computadores, esses equipamentos eram tão grandes que deveriam ser instalados em salas enormes. Essa foi a era dos *mainframes*, na qual várias pessoas compartilhavam um único computador. Com o passar dos anos e a difusão cada dia maior da tecnologia, esses dispositivos foram diminuindo de tamanho, até chegarem à arquitetura dos *Personal Computers* (PCs); essa pode ser vista como a era da *computação pessoal*, na qual uma pessoa utiliza um computador. Após isso, com a evolução da Internet e da computação distribuída, surge uma nova era, a era da *computação ubíqua*, em que vários computadores são compartilhados para um único usuário [Weiser e Brown 1996].

Mark Weiser definiu o termo computação ubíqua como a capacidade de integrar recursos computacionais no cotidiano das pessoas de forma transparente e para os mais diversos fins [Weiser 1991]. A computação ubíqua não se limita somente em implantar computadores nos ambientes, mas sim, na conexão de vários dispositivos para a troca de informações entre si, a fim de auxiliar na vida do usuário. Esses recursos são integrados no ambiente para auxiliar os usuários na realização de suas tarefas, mesmo que nunca tenham utilizado um computador antes. Com a computação ubíqua, o foco dos usuários não é voltado para a ferramenta que estão utilizando, mas sim, para a tarefa que estão realizando [Weiser 1994].

Weiser previa que diferentes dispositivos, de diferentes tamanhos (pequenos, médios e grandes) e diferentes finalidades (para uso pessoal e coletivo) surgiriam com o passar dos anos. De fato, isso acabou acontecendo: dispositivos que antes eram muito raros, como por exemplo, *notebooks*, no final da década de 90 já estavam disponíveis comercialmente. Pôde-se perceber o crescimento da utilização desses dispositivos, desde o uso pessoal (*laptops*, *Tablet PCs*, *personal digital assistants* etc) a uso coletivo (lousa eletrônica, *mimios*¹ etc). Alguns dispositivos de computação ubíqua são ilustrados na Figura 2.1.

Computação ubíqua é um tema de pesquisa em Ciência da Computação que engloba várias de suas disciplinas, como Computação Móvel, Sistemas Distribuídos, Interação Humano-Computador, entre outras. Essas áreas estão diretamente relacionadas, muitos problemas na computação ubíqua são resolvidos utilizando conceitos e soluções propostas por elas. A união da computação ubíqua com essas áreas proporciona grandes ganhos, criando novos problemas e soluções para serem aplicados na computação.

2.2.1 Classificação das Aplicações de Computação Ubíqua

[Abowd e Mynatt 2000] consideram que as aplicações de computação ubíqua podem ser divididas em três áreas de pesquisa: interfaces naturais, computação ciente de contexto

¹ *Mimio* é um dispositivo capaz de rastrear sinais com as coordenadas de posicionamento emitidas por uma caneta eletrônica. A superfície é mapeada para a projeção da área de trabalho do computador ao qual o *mimio* está conectado, fazendo com que a caneta eletrônica funcione como um mouse, provendo à superfície características semelhantes a uma lousa. (<http://www.mimio.dymo.com/>)



Figura 2.1: Dispositivos de computação ubíqua (lousa eletrônica, *Tablet PCs*, caneta eletrônica).

e captura e acesso de atividades humanas. Cada uma dessas áreas será explicada em mais detalhes a seguir:

Interfaces Naturais

Um dos grandes objetivos da computação ubíqua é alterar a forma como os usuários interagem com os recursos computacionais. As interfaces naturais procuram quebrar o paradigma de que o computador é composto por componentes tradicionais, como *mouse*, teclado e *display*, e buscam integrar novas formas de interação entre os humanos e os computadores. Nesta óptica, diversas pesquisas focam em interfaces menos intrusivas, como interação baseada em voz, escrita e gestos, integração de sensores ao ambiente, *display* sensíveis ao toque, entre diversas outras abordagens.

As pesquisas concentradas em interfaces naturais estão em contínuo crescimento, a cada dia surgem novas tecnologias para deixar a interação do usuário com o computador mais transparente. Dispositivos de videogame estão explorando fortemente os conceitos de interfaces naturais, propondo formas de interação que visam a dar maior realidade ao jogo, como é o caso do *Nintendo Wii*² e do *XBox 360 + Kinect*³.

Computação ciente de contexto

Essa classe de aplicações baseia-se em informações do contexto para se adaptar ao ambiente no qual opera. A ideia é que os dispositivos possam identificar e reagir ao ambiente em que estão inseridos com o intuito de facilitar a realização da tarefa pelo usuário. Contexto pode ser visto como qualquer informação relevante sobre as entidades (pessoa, lugar, objeto) que participem da interação [Dey 2001].

De acordo com [Cattelan 2009], aplicações de computação ubíqua precisam ter ciência

²<http://www.nintendo.com/wii>

³<http://www.xbox.com/pt-BR/Kinect>

de contexto, de modo a serem capazes de adaptar seu comportamento com base nas informações extraídas dos ambientes físico e computacional. [Hess e Campbell 2003] afirmam que um dos fatores que diferenciam a computação ubíqua da computação tradicional é o contexto, que permite ao sistema se adequar ao ambiente para facilitar o uso dos recursos computacionais. As tecnologias de ciência de contexto podem ser aplicadas nos mais diversos níveis de profundidade, desde a utilização em uma simples identificação de usuário, até mesmo a complexos mecanismos para classificação e recomendação de conteúdo ao usuário.

Captura e acesso de atividades humanas

Um dos potenciais empregos de computação ubíqua é a capacidade de integrar recursos computacionais no ambiente, a fim de propiciar que experiências vividas pelos usuários sejam registradas e disponibilizadas, posteriormente, para acesso. Por essas experiências produzirem grandes volumes de dados, dos mais diversos formatos, tornam-se necessárias sofisticadas aplicações para gravar, sincronizar e gerar documentos derivados a partir dessas experiências. As aplicações de C&A surgem para dar suporte ao processo de capturar e disponibilizar informações produzidas pelos usuários durante períodos de sua vida.

As aplicações de C&A integram diferentes dispositivos de captura, como câmeras, microfones, sensores, entre outros, para capturar tudo que acontece em um determinado local, possibilitando que os usuários se concentrem na experiência em si, sem a necessidade de registrar a informação manualmente, pois tudo que está acontecendo será capturado automaticamente pela aplicação. Por ser um tema de grande relevância para o trabalho, mais detalhes de aplicações de C&A serão apresentados na Seção 2.3.

Embora C&A sejam o foco deste trabalho, este também explora aspectos relacionados aos temas de interfaces naturais e ciência de contexto.

2.3 Captura e Acesso

A ideia de registrar informações do dia a dia das pessoas não é uma proposta recente. Em 1945, [Bush e Wang 1945] escreveram sobre os benefícios de um sistema computacional generalizado para C&A de informações. Na sua proposta, o *memex* seria um sistema destinado a armazenar todos os artefatos com os quais uma pessoa entraria em contato em uma determinada ocasião e todas as associações criadas entre eles. Após isso, pesquisadores procuraram desenvolver mais pesquisas com o intuito de validar a abordagem de C&A.

Captura e acesso podem ser definidas como as tarefas de registrar informações de uma experiência ao vivo e permitir que estas possam ser, posteriormente, acessadas pelos usuários [Truong et al. 2001]. A captura é o ato de registrar as informações de uma deter-

minada atividade, gerando documentos passíveis de armazenamento. Esses documentos podem ser dos mais variados formatos. Já o acesso é o ato de recuperar ou revisar as informações geradas a partir de uma atividade de captura. Essa etapa deve apresentar as informações de forma organizada, possibilitando que o usuário tenha acesso ao conteúdo de forma clara.

As atividades de C&A não envolvem somente o processo de capturar e disponibilizar as informações, mas sim, a uma série de atividades entre essas duas etapas. A etapa de armazenamento, por exemplo, é uma importante atividade, em que as informações devem ser armazenadas seguindo algum tipo de estruturação para facilitar o processo de recuperação da informação. A atividade de extensão pode envolver desde o reúso das informações até a criação de ligações (*links*) hipermídia entre documentos com conteúdo relacionado.

Neste sentido, os sistemas que tratam a abordagem de C&A têm sido explorados em diversos domínios, cada um com suas particularidades e características.

2.4 Questões de Projeto de Aplicações de Captura e Acesso

As aplicações de C&A, por sua natureza, impõem uma série de características que são importantes para o seu processo de desenvolvimento.

2.4.1 Dimensões dos Projetos de Captura e Acesso

[Truong e Hayes 2009] mostram que projetos de aplicações de C&A podem ser descritos por meio de cinco dimensões:

- A primeira dimensão refere-se a quando e onde a captura ocorre. Essa determina aspectos referentes à forma como os dados serão capturados: se a captura ocorre de forma fixa, móvel ou vestível;
- A segunda dimensão determina como a tarefa de captura será ativada e como as anotações serão associadas com os dados em si;
- A terceira dimensão refere-se ao número de dispositivos que farão parte do processo, como: quantos e quais os tipos de dispositivos de captura são associados com a aplicação de captura, quantos e quais tipos de dispositivos de armazenamentos também serão associados com a aplicação;
- A quarta dimensão diz respeito às técnicas utilizadas para visitar as informações capturadas, pois esta determina a forma como os usuários acessarão os dados;

- A quinta dimensão refere-se ao tempo em que as informações capturadas ficarão disponíveis para o usuário, por quanto tempo os dados persistirão.

2.4.2 Domínio e Heterogeneidade

As aplicações de C&A podem estar dispostas nos mais variados domínios de aplicação. Pode-se perceber uma concentração maior em projetos de captura de reuniões e aulas, pois estes, se adaptam melhor à proposta e às características dessas classes de aplicações.

No processo de captura de informações do ambiente instrumentado, uma série de recursos, dos mais variados gêneros, são utilizados. Segundo [Cattelan 2009], a captura ocorre em uma espécie de eixo vertical, envolvendo desde dispositivos de hardware, como microfones e câmeras, até sofisticados componentes de software.

2.4.3 Privacidade das Informações

Aplicações de C&A geram uma grande quantidade de informações que poderão ser apresentadas aos usuários posteriormente. A forma como essas informações serão disponibilizadas envolve uma série de fatores, como, por exemplo, questões de privacidade do conteúdo. Esses registros para acesso futuro podem acontecer em dois níveis: acesso público, em que o conteúdo capturado consiste de artefatos públicos — criados por membros de um determinado grupo, sendo proporcionado acesso a todos desse grupo — e acesso privado, em que o conteúdo capturado consiste de artefatos privados — criados, manipulados e pertencentes somente a uma pessoa [Greenberg et al. 1999].

Há riscos das informações serem compartilhadas de forma não autorizada, [Weiser 1993] rejeita a ideia de um repositório central para armazenamento de informações privadas, defendendo que a informação particular do usuário deve ser mantida em seu dispositivo pessoal.

2.4.4 Captura e Acesso Embarcada

A abordagem de C&A muitas vezes foca nas características de ubiquidade somente na etapa de captura, deixando o processo de acesso não tão transparente ao usuário. Com isso, [Kientz 2012] propõe uma nova abordagem para as aplicações de C&A: a captura e acesso embarcada, que tem a característica de permitir que tanto o processo de captura, quanto o processo de acesso, seja totalmente, ou quase totalmente, transparente ao usuário. Essa proposta é bastante relevante, visto que, não raro, a fase de acesso não é tão levada em consideração quanto a fase de captura.

A captura e acesso embarcada expande a noção tradicional de C&A definida por [Abowd e Mynatt 2000]. Essa nova abordagem busca incentivar os usuários a realizarem mais revisão do conteúdo capturado, não deixando que esse processo gere enormes quan-

tidades de dados que não sejam visualizadas. Se o processo de acesso às informações for mais transparente e ubíquo ao usuário, ele ficará mais motivado a interagir com sistemas desse propósito [Kientz 2012].

2.5 Ambientes de Captura e Acesso

Aplicações de C&A são dispostas nos mais variados domínios de aplicação para apoiar a realização das mais diversas atividades. Segundo [Truong 2005], os domínios para essas aplicações se dividem em três: sala de aula, sala de reuniões e ambientes de propósitos gerais. Nas próximas seções, serão descritas as características de cada domínio e trabalhos reportados na literatura.

2.5.1 Captura e Acesso em Salas de aula

A C&A com propósitos educacionais visam integrar dispositivos de captura, como lousa eletrônica, microfones, câmeras, entre outros, para propiciar a captura automática de todas as informações ministradas em uma sala de aula. O processo de captura deve acontecer de forma transparente ao professor, permitindo que ele interaja com o sistema sem alterar a forma de condução da aula. Geralmente, após uma sessão de captura os fluxos de informações são sincronizados para gerar documentos que permitam o acesso pelos alunos.

A tarefa de registrar informações em sala de aula vem sendo explorada há algum tempo, desde quando muitas contribuições para a comunidade científica puderam ser alcançadas. Dentre esses projetos que tiveram sucesso, pode-se citar o *eClass* [Abowd 1999] (conhecido como *Classroom 2000*) o qual tem o objetivo de gravar uma série de informações de uma aula. As apresentações são convertidas automaticamente em *slides* projetados sobre uma lousa eletrônica onde anotações podem ser feitas de forma manual. Um microfone é utilizado para capturar tudo o que o instrutor diz, da mesma forma que uma câmera é usada para capturar uma visão fixa da sala de aula. O sistema proporciona também, o registro de todo o conteúdo Web visitado durante uma sessão. Uma extensão do *eClass* pode ser vista em [Pimentel et al. 2007].

Alguns sistemas contam com câmeras e técnicas de visão computacional para capturar o material escrito e exposto na lousa. Câmeras adicionais podem ser empregadas para prover diferentes imagens ou vídeos da sala de aula [Black et al. 1998]. “*Cornell Lecture Browser*” [Mukhopadhyay e Smith 1999] é um sistema que estrutura a interface de apresentação das informações. As interfaces usam uma *timeline* (linha do tempo) que facilita a navegação temporal das aulas. Outro sistema bem consolidado, descrito na literatura, é o *E-Chalk* [Friedland et al. 2004], um sistema de ensino que utiliza projeções para criar a noção de uma grande lousa eletrônica. Todas as informações são armazenadas em um

servidor que permite a transmissão de todo conteúdo capturado pela internet.

Muitas aplicações de C&A focam na construção de arquiteturas com suporte à colaboração de conteúdo. *Student Notepad (StuPad)* [Truong et al. 1999] é um sistema que foi adicionado ao *eClass* para fornecer aos estudantes uma interface capaz de integrar o que está sendo ministrado pelo professor com o computador pessoal de cada aluno. Todas as anotações realizadas pelos alunos são capturadas e disponibilizadas na Web para facilitar sua revisão a qualquer hora e lugar. *DEBBIE* [Berque et al. 1999] é um sistema muito semelhante ao *StuPad*, pelo qual o instrutor apresenta o conteúdo em uma lousa eletrônica, e as informações são capturadas e repassadas para os *notebooks* dos alunos. Ambos os sistemas possuem anotações públicas e privadas, mas somente o *StuPad* permite que as anotações sejam escritas sobre o slide do professor [Truong e Hayes 2009].

Diversos sistemas são voltados para usuários com algum tipo de limitação, por exemplo, em [Giliberti et al. 2011] é exposto um sistema para auxiliar pessoas com necessidades educacionais especiais. É um sistema baseado em lousa e *Tablet PCs* para registro de áudio e vídeo de aulas. Todos os fluxos são armazenados em um servidor que armazena as informações a serem disponibilizadas aos alunos. Outro exemplo é o “*The Walden System*” [White et al. 2003], proposto para capturar o comportamento de crianças com autismo. O objetivo principal deste trabalho é investigar o desenvolvimento da interação do usuário com outras crianças e com o professor.

2.5.2 Captura e Acesso em Salas de Reuniões

As aplicações de C&A também têm sido investigadas no domínio de sala de reuniões. Geralmente, esses ambientes requerem que os participantes realizem muitas anotações sobre as informações que são passadas durante esses eventos. Com isso, faz-se importante a implantação de um sistema que capture automaticamente todas as informações com a possibilidade de disponibilização posterior aos participantes.

Diversos sistemas foram propostos com o passar dos anos. Um exemplo que teve muito sucesso pode ser visto em [Pedersen et al. 1993], uma aplicação baseada em lousa eletrônica para permitir a colaboração entre pequenos grupos de participantes que capturam as anotações realizadas por eles. Buscando alcançar suporte a reuniões distribuídas, o *DOLPHIN* [Streitz et al. 1994] foi projetado para que usuários pudessem interagir em uma lousa eletrônica compartilhada.

O *LiteMinutes* [Chiu et al. 2001] é um sistema capaz de capturar de forma síncrona vários fluxos de informações, como áudio e vídeo de uma reunião. A sincronização acontece por meio de marcação de tempo fixada durante o processo de captura. As mídias capturadas são combinadas em um documento HTML com referências para as outras mídias. Após a captura, o conteúdo pode ser acessado pela Web. O *FiloChat* [Whittaker et al. 1994] utiliza dispositivos móveis, como *Tablets PCs*, para apoiar a captura das

informações durante uma reunião. Além disso, o *FiloChat* disponibiliza um recurso que possibilita que as sessões capturadas sejam organizadas em seções.

Outro importante sistema disponível na literatura é o *SAAMpad* [Richter et al. 1998], um sistema voltado para sessões de discussão sobre arquitetura de software. As informações capturadas são áudio, vídeo e diagramas feitos sobre lousas eletrônicas. A aplicação dá suporte a desenhos de objetos específicos, como diagramas, fluxos de dados, entre outros.

2.5.3 Captura e Acesso para Propósitos Gerais

Muitas pesquisas em C&A são desenvolvidas além de abordagens em sala de aulas e reuniões. Mesmo sendo menos numerosas, muitas dessas aplicações têm sucesso e são empregadas em ambientes reais.

Uma das primeiras pesquisas fora do domínio de aulas e reuniões foi a aplicação *Forget-Me-Not* [Lamming e Flynn 1994]. Essa foi a primeira aplicação a demonstrar o conceito de captura contínua de experiências de usuário em ambiente de captura instrumentado. Outra proposta é o *SenseCam* [Hodges et al. 2006], dispositivo de captura passiva, móvel e pessoal, que registra automaticamente imagens da vida das pessoas. Esse dispositivo é integrado com câmeras, acelerômetro e sensores de temperatura, luz e movimento.

Alguns sistemas de C&A são aplicados a domínios como conferências e visitas a museus. O *Conference Assistant* [Dey et al. 1999] é um sistema que permite aos usuários realizarem anotações de acontecimentos a partir de dispositivos móveis. Todos os dados capturados são integrados com informações de sensores dispostos no ambiente. O *HP Remember* [Fleck et al. 2002] é um sistema que auxilia visitantes de museu a documentar todas as informações obtidas durante a visita. Esse sistema gera uma página Web automaticamente com parte do conteúdo fornecido pelo usuário e outra parte fornecida por sensores disponíveis no ambiente.

Outro domínio em que se aplica muito o conceito de C&A pode ser visto em ambientes médicos. Em [Mamykina et al. 2004], é apresentado um sistema para dar suporte a atividades de cuidados domiciliares por enfermeiros. Em [Plaisant et al. 1996], é proposto um sistema que foca na visualização de dados de pacientes em longos períodos de tempo. Outra abordagem pode ser vista em [Xu et al. 2006], um sistema para apoiar capturas de sessões de fisioterapia em pacientes vítimas de AVC (Acidente Vascular Cerebral). Esse sistema auxilia nas anotações colaborativas em tempo real e visualização dessas atividades, produzindo registros interativos do histórico da reabilitação.

2.6 Características Comuns às Aplicações de Captura e Acesso

Segundo [Truong e Hayes 2009], existem quatro requisitos não funcionais impostos para as aplicações de C&A:

- **Captura Deve Acontecer Naturalmente:** Aplicações de C&A devem auxiliar usuários na realização de suas tarefas de modo que eles não desviem sua atenção da experiência apresentada. Para isso, o ambiente de captura deve ser estruturado de forma não intrusiva, e deve exigir nenhum ou pouco esforço adicional dos usuários para utilizarem o sistema.
- **Informações Devem ser Acessíveis com Mínimo Esforço:** O projeto de aplicações de C&A envolve bem mais que o desenvolvimento de serviços de captura discretos; a fase de acesso deve apresentar-se de forma acessível ao usuário. O acesso é mais útil quando ele está disponível por toda parte, ou seja, usuários necessitam de interfaces que permitam o acesso em qualquer lugar e a qualquer hora.
- **Registros Devem ser Indexados por Conteúdo e Contexto:** Com um certo tempo, a captura automática resulta em grandes quantidades de informações. Esse grande conteúdo de dados pode dificultar a tarefa de busca por uma determinada informação. Essas interfaces de acesso devem oferecer aos usuários formas claras e organizadas para acesso às informações. Índices — como títulos, palavra-chaves e resumo — podem ser criados para que usuários possam facilmente encontrar as informações desejadas.
- **Registros Devem ser Criados, Armazenados e Acessados de Modo Ético e Legal:** Atividades de C&A podem armazenar uma grande quantidade de informações pessoais do dia a dia dos usuários. Os sistemas de armazenamento e apresentação do conteúdo devem ser capazes de garantir políticas distintas para o acesso a conteúdos públicos e privados. As informações podem ser classificadas de acordo com sua condição de privacidade.

2.7 Considerações Finais

Como pôde ser visto, computação ubíqua é uma importante área de pesquisa em Ciência da Computação. Suas aplicações contribuem de forma significativa na vida dos seus usuários. Unidas ao conceito de computação ubíqua, as aplicações de C&A também são de grande relevância para a computação, por produzirem artefatos de software que auxiliam usuários na realização de tarefas do dia a dia. Com as aplicações de C&A, é possível capturar diversos fluxos de informações disponíveis no ambiente exibindo-as, posteriormente, ao usuário.

Neste capítulo, foram tratados conceitos referentes à computação ubíqua e à área de pesquisa em C&A de conteúdo multimídia. Várias características dessas aplicações foram discutidas. Foram apresentados os domínios de aplicação e diversos trabalhos encontrados na literatura que abordam pesquisas em C&A. Por fim, foram descritos alguns aspectos que devem ser considerados no processo de desenvolvimento dessas aplicações.

Capítulo 3

Classroom eXperience

3.1 Considerações Iniciais

O desenvolvimento de aplicações de captura e acesso (C&A) demanda a implementação de complexos componentes de software que capturam as informações do ambiente e as disponibilizam, posteriormente, aos usuários. Esses componentes de software devem ser bem estruturados, a fim de permitir uma sequência organizada no processo gravação, extensão, armazenamento e acesso aos dados.

Este capítulo apresenta o *Classroom eXperience* (CX), um sistema que dá suporte à captura de atividades ao vivo em ambientes educacionais instrumentados. Na Seção 3.2 são apresentados a arquitetura geral do sistema, o relacionamento entre os seus principais componentes e os detalhes de implementação. Na Seção 3.3 é mostrado o funcionamento do mecanismo para tratamento contextual que recomenda informações aos usuários. Na Seção 3.4 é mostrado o comportamento do sistema, em que pode ser vista a forma pela qual cada etapa se relaciona às demais. Por fim, na Seção 3.5 são apresentadas as considerações finais sobre o capítulo.

3.2 Arquitetura do *Classroom eXperience*

O *Classroom eXperience* (CX) é um sistema fundamentado em computação ubíqua, que foi projetado para dar suporte a atividades de captura e apresentação de conteúdo provenientes de ambientes educacionais instrumentados. Esse sistema é composto por diversos componentes de hardware e software, os quais permitem que todas as informações apresentadas por um professor durante uma aula sejam capturadas e disponibilizadas, posteriormente, aos alunos. O CX possibilita que professores ministrem aulas utilizando recursos computacionais sem alterar sua dinâmica convencional. O sistema proposto auxilia os professores na realização de suas atividades, bem como os alunos no acesso às informações capturadas.

As atividades de captura acontecem em uma sala de aula instrumentada com dispositivos que registram todas as informações durante a aula. Assim, câmeras, microfones, lousas eletrônicas e projetores são integrados ao ambiente para proporcionar tal funcionalidade. A captura ocorre de forma contínua durante a aula, todos os fluxos de informações são capturados por meio de componentes de software que são associados aos respectivos dispositivos de captura (câmera, microfone e lousa eletrônica).

A ideia é que, inicialmente, o professor cadastre a aula a ser ministrada por meio de interfaces Web disponibilizadas pelo sistema. No momento do cadastro, o professor informa todos os dados e metainformações sobre a aula em questão. Quando chegar o momento da aula, o professor se dirige à sala para iniciar a sessão de captura. Chegando na sala instrumentada, o professor acessa sua área no sistema e seleciona qual aula quer capturar. Nesse momento, uma aplicação cliente, parte do CX, responsável por realizar a captura das informações, é transferida para o computador do professor. Desse momento em diante todos os fluxos de informações passam a ser capturados por meio de componentes de software integrados à infraestrutura. Após o término da aula, o professor encerra o processo de captura no sistema. A partir daí, as informações são sincronizadas pelos componentes de sincronização e são gerados documentos multimídia derivados para apresentação ao usuário em formatos HTML e NCL¹. Depois da geração dos documentos, estes são transferidos para serviços de armazenamento na rede por meio de um protocolo, também desenvolvido no contexto deste trabalho. Assim que a transferência é realizada, as informações estarão disponíveis em uma plataforma Web para possíveis revisões pelos alunos.

Para o desenvolvimento dos ambientes Web, foram utilizadas tecnologias disponibilizadas pela linguagem Java. O PostgreSQL² foi o Sistema de Banco de Dados utilizado para o armazenamento dos dados. Foi modelada uma base de dados relacional que possui um total de 18 tabelas, em que se destacam as tabelas de usuários, disciplinas, turmas e aulas. O Apêndice A mostra um diagrama do Modelo Entidade-Relacionamento para o banco de dados utilizado no CX.

As aplicações de C&A, geralmente, seguem uma sequência bem estruturada de atividades que devem acontecer entre a captura das informações e a sua posterior disponibilização. Os problemas de C&A devem ser divididos em quatro fases: Pré-produção, Gravação ao vivo, Pós-produção e Acesso [Abowd et al. 1996]:

- **Pré-produção:** Consiste na preparação do conteúdo para a fase de captura;
- **Gravação ao vivo:** Consiste na captura dos fluxos de informações apresentadas durante uma aula;

¹NCL [Silva et al. 2004] (*Nested Context Language*) é o padrão do novo Sistema Brasileiro de TV Digital (SBTVD) para programas declarativos interativos, promovido a padrão internacional ITU-T (especificação H.761), e permite especificar, de forma estruturada, documentos hipermídia complexos contendo relacionamentos de sincronização temporal e espacial entre seus componentes.

²<http://www.postgresql.org/>

- **Pós-produção:** Responsável por sincronizar os fluxos de informações capturados e gerar documentos multimídia em formatos que possam ser apresentados aos alunos;
- **Acesso:** Consiste na apresentação do conteúdo previamente capturado ao usuário final.

A Figura 3.1 exibe a estruturação das fases do problema, a mesma adotada na infraestrutura do CX.



Figura 3.1: As quatro fases no processo de captura e acesso.

Nas subseções a seguir, serão detalhadas cada umas das fases citadas, e também serão expostos aspectos de concepção e implementação dos artefatos de software utilizados em cada etapa.

3.2.1 Pré-produção

No CX, a etapa de pré-produção consiste na configuração do conteúdo que será capturado durante uma aula. Nesta etapa, o professor irá cadastrar todas as informações pertinentes ao conteúdo da aula mediante uma interface *frontend* disponível na Web. As informações são organizadas em 3 categorias: dados básicos da aula, dados adicionais (metainformações) e o conteúdo a ser apresentado.

A primeira categoria, dados básicos, são informações diretamente relacionadas à aula, como: título da aula, duração e palavras-chave. Esses dados são armazenados em banco de dados relacional contendo ligações que associam este conteúdo com as demais informações da aula.

As metainformações, segunda categoria, são informações secundárias sobre a aula, como: tipo de interatividade, tempo de aprendizado, recurso de aprendizado, entre outros. As metainformações da aula são classificadas e organizadas segundo extensões criadas a partir de modelo de metainformações para objetos de aprendizagem, o IEEE-LOM [LTSC

IEEE 2002]. O IEEE-LOM é um modelo de dados, geralmente codificado em XML, que especifica a sintaxe e a semântica de metadados para objetos de aprendizado. As metainformações cadastradas pelo professor são armazenadas em um modelo de dados que foi simplificado do padrão LOM. A utilização de metadados que estendem objetos de aprendizado, pode permitir a integração da infraestrutura com outros ambientes que suportam recursos de aprendizado.

Na terceira e última categoria, as informações a serem apresentadas aos alunos são os dados que serão projetados sobre a lousa eletrônica. Esses dados são recebidos por componentes de transformação que estenderão esse conteúdo para um formato reconhecido pelo sistema de captura. O CX deve transformar o conteúdo enviado pelo professor em um formato padrão que o sistema reconheça, posteriormente, na etapa de captura. O professor pode enviar o arquivo a ser apresentado em diversos formatos, sendo que o componente de transformação deve padronizar esse conteúdo em um formato que seja reconhecido pelo componente de gravação.

Após o processamento de todos os dados pela etapa de pré-produção, o sistema já está pronto para iniciar um processo de captura (gravação ao vivo).

3.2.2 Gravação ao vivo

A fase de gravação ao vivo consiste no processo de capturar as informações que serão ministradas por um professor durante uma aula. A partir do conteúdo já cadastrado, o professor acessa uma interface Web e recebe o conteúdo a ser apresentado para o aluno como foi cadastrado anteriormente. Uma aplicação cliente, parte do sistema CX, recebe as informações cadastradas pelo professor e inicia todos os componentes que são responsáveis pelo registro da “sessão”. Uma “sessão” de captura pode ser definida como o período de interação entre os participantes de uma experiência ao vivo, durante o qual ocorre produção de informação [Cattelan 2009].

A aplicação cliente é executada por meio de uma tecnologia disponibilizada pela linguagem Java denominada JWS³. Essa tecnologia permite aos usuários fazer o download de aplicativos Java e executá-los. JWS elimina procedimentos complexos de instalação e atualização de aplicativos, pois garante que sempre a última versão está sendo executada.

Essa etapa de gravação ao vivo é composta por três componentes de software, cuja função é capturar as informações que serão transmitidas pelo professor: o componente de captura de áudio, o componente de captura de vídeo e o componente de captura das anotações feitas sobre a lousa eletrônica. Esses componentes são integrados com os seus respectivos componentes de hardware para registrar todas as informações apresentadas, utilizando chamadas de serviços para armazenamento local das informações que estão sendo capturadas. Todos os fluxos de informação são armazenados localmente durante o

³JWS (*Java Web Start*) - http://www.java.com/pt_BR/download/faq/java_webstart.xml

processo de captura.

Um dos desafios de projetar aplicações para C&A é fazer com que a etapa de captura aconteça de forma transparente ao usuário. O processo de captura busca ser o menos intrusivo possível, permitindo que o professor não altere a forma como ministra sua aula. A Figura 3.2 ilustra uma aula sendo capturada utilizando o CX. O professor utiliza os recursos do sistema para interagir com a lousa eletrônica como se estivesse utilizando uma lousa tradicional. Todas as informações escritas sobre a lousa eletrônica são projetadas em tela para facilitar a visualização pelos alunos.

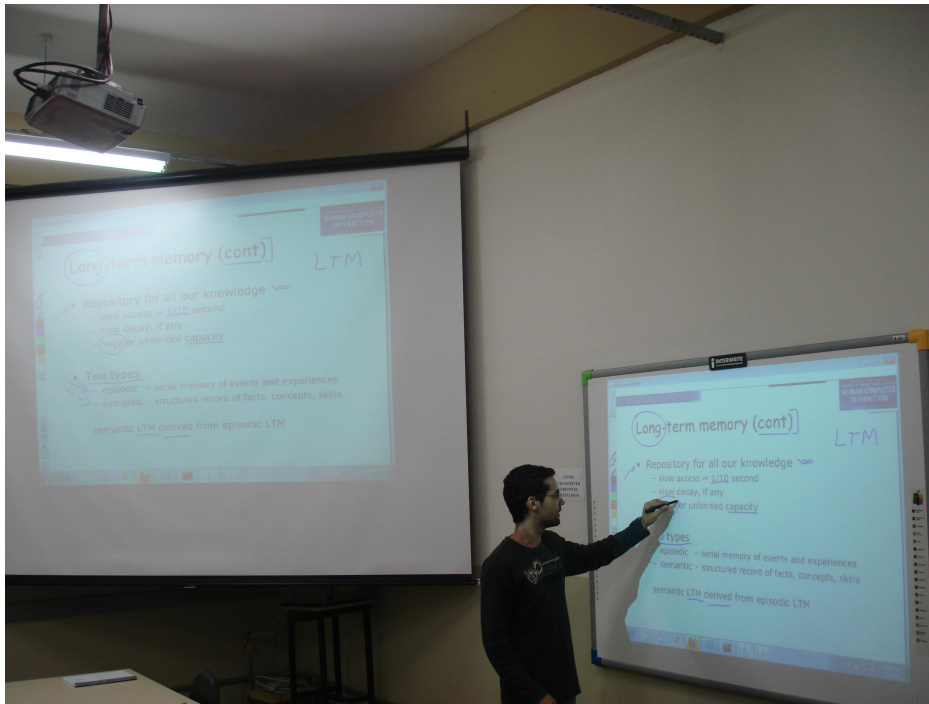


Figura 3.2: Aula sendo ministrada com recursos do *Classroom eXperience*.

Após a etapa de gravação de todas as informações passadas pelo professor, o sistema deve sincronizar as mídias geradas e criar documentos que sejam acessíveis aos usuários. Esse processo é realizado na etapa de pós-produção, apresentada a seguir.

3.2.3 Pós-produção

A etapa de pós-produção se inicia logo após o término da etapa de captura. Esta etapa é responsável pela sincronização dos fluxos de mídia capturados e pela geração de documentos a serem apresentados para o usuário. Todas as informações previamente capturadas são sincronizadas por meio de documentos XML, que descrevem toda a interação entre o professor e o sistema de captura. É gerado um documento XML para cada aula capturada, que, por sua vez, possui referências para cada elemento capturado (vídeo, áudio e anotações sobre a lousa eletrônica). A API JAXB⁴, provida pela Sun Microsys-

⁴JAXB (*Java Architecture for XML Binding*) - <http://jaxb.java.net/>

tems, foi utilizada para fazer o mapeamento entre objetos Java e documentos XML. Essa API gera classes Java a partir de esquemas XML de cada componente responsável pela captura. Foram utilizados os mesmos esquemas disponibilizados pelo *iClass*.

O subsistema responsável pela etapa de pós-produção pode ser resumido em três componentes:

- **O componente de sincronização:** Esse componente recebe as informações capturadas e faz a sincronização por meio de documentos XML. Esse documento contém informações referentes ao contexto da aplicação, bem como as informações sobre os dispositivos de captura. Exemplo de um documento XML gerado pelo componente de sincronização é ilustrado na Figura 3.3.

```
- <session id="1">
- <app_context>
  <person>35</person>
  <place>1B126</place>
  <term>2012-1</term>
  <course_name>Interação Humano-Computador</course_name>
  <class_title>Aula 5 - O Computador (parte2)</class_title>
  <cod_class>251</cod_class>
</app_context>
- <whiteboardml board_width="1228" board_height="710">
+ <slide slide_id="0">
+ <slide slide_id="1">
+ <slide slide_id="2">
+ <slide slide_id="3">
+ <slide slide_id="4">
- <visit slide_id="0" duration="11062" visit_id="0" timestamp="0">
  - <stroke board_height="710" board_width="1228" timestamp="1859" person_id="35"
    width="1" stroke_id="1" color="java.awt.Color[r=0,g=0,b=255]" timestamp_end="5125">
    <min_x>180</min_x>
    <min_y>172</min_y>
    <max_x>788</max_x>
    <max_y>482</max_y>
    <points>
      {230.0,275.0,224.0,275.0,220.0,276.0,213.0,279.0,205.0,285.0,198.0,293.0,...
    </points>
  </stroke>
  + <stroke board_height="710" board_width="1228" timestamp="6391" person_id="35"
    width="1" stroke_id="2" color="java.awt.Color[r=0,g=0,b=255]" timestamp_end="8719">
  </visit>
+ <visit slide_id="1" duration="4313" visit_id="1" timestamp="11062">
+ <visit slide_id="2" duration="4000" visit_id="2" timestamp="15375">
</whiteboardml>
<audioml />
<videoml />
</session>
```

Figura 3.3: Documento XML gerado pelo componente de sincronização.

- **O componente de transformação:** A função desse componente é criar extensões para a apresentação ao usuário a partir do conteúdo capturado. Esse componente recebe dois tipos de elementos: os documentos XML gerados a partir do componente de sincronização e folhas de estilo XSLT⁵, que determinam as regras para a trans-

⁵XSLT (*eXtensible Stylesheet Language for Transformation*) - www.w3.org/TR/xslt

formação. O componente de transformação cria extensões do conteúdo capturado através da geração de documentos em formatos HTML e NCL, para apresentação na plataforma de TV Digital Interativa. NCL está sendo utilizada como a linguagem padrão do middleware GINGA empregado no sistema brasileiro de TV Digital. A Figura 3.4 mostra exemplo de trecho de um documento NCL gerado pelo componente de transformação.

```
- <ncl id="session" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
- <head>
  <connectorBase />
  + <regionBase>
  + <descriptorBase>
</head>
- <body>
  <port id="pInicio" component="audio" />
  <media type="image/jpeg" id="fundo" src="fundo.JPG" descriptor="dFundo" />
  <media type="image/jpeg" id="icon_audio_v" src="icon_audio_v.JPG" descriptor="dicon_audio" />
  <media type="image/jpeg" id="icon_audio_x" src="icon_audio_x.JPG" descriptor="dicon_audio" />
  <media type="image/jpeg" id="icon_video_v" src="icon_video_v.JPG" descriptor="dicon_video" />
  <media type="image/jpeg" id="icon_video_x" src="icon_video_x.JPG" descriptor="dicon_video" />
  <media type="image/jpeg" id="icon_prev" src="icon_prev.JPG" descriptor="dicon_prev" />
  <media type="image/jpeg" id="icon_next" src="icon_next.JPG" descriptor="dicon_next" />
  <media type="image/jpeg" id="slide0" src="slide0.JPG" descriptor="dSlide" />
  <media type="image/jpeg" id="slide1" src="slide1.JPG" descriptor="dSlide" />
  <media type="image/jpeg" id="slide2" src="slide2.JPG" descriptor="dSlide" />
  <media type="image/jpeg" id="slide3" src="slide3.JPG" descriptor="dSlide" />
  + <media id="video" type="video/mpeg" src="video.wmv" descriptor="dVideo">
  + <link id="Inicio" xconnector="onBeginStart">
  + <link id="Slide0_1" xconnector="onBeginStartStop">
  + <link id="Slide0_next" xconnector="onKeySelectionStopStart">
  + <link id="Slide1_prev" xconnector="onKeySelectionStopStart">
  + <link id="Slide1_2" xconnector="onBeginStartStop">
  + <link id="Slide1_next" xconnector="onKeySelectionStopStart">
  + <link id="Slide2_prev" xconnector="onKeySelectionStopStart">
  + <link id="Slide2_3" xconnector="onBeginStartStop">
  + <link id="Slide2_next" xconnector="onKeySelectionStopStart">
  + <link id="Slide3_prev" xconnector="onKeySelectionStopStart">
  + <link id="StopVideo" xconnector="onKeySelectionSetStopStart">
  + <link id="StartVideo" xconnector="onKeySelectionSetStopStart">
  + <link id="StopAudio" xconnector="onKeySelectionSetStopStart">
  + <link id="StartAudio" xconnector="onKeySelectionSetStopStart">
</body>
</ncl>
```

Figura 3.4: Trecho de um documento NCL gerado pelo componente de transformação.

- **O componente de compactação:** Esse componente é caracterizado por produzir um documento que seja reconhecido pela etapa de acesso. Esse procedimento consiste em agrupar todas as informações produzidas em um arquivo compactado.

3.2.4 Acesso

A etapa de acesso consiste na apresentação ao usuário dos documentos gerados pela etapa de pós-produção. Todos os fluxos que foram capturados durante a aula são expostos ao aluno de forma clara e organizada. Este conteúdo poderá ser acessado pela Web ou pela TV Digital interativa por meio do Sistema Brasileiro de Televisão Digital (SBTVD).

Segundo [Kientz 2012], o processo de acesso nas aplicações de C&A muitas vezes é deixado como etapa secundária, não sendo empregado o esforço necessário para o seu desenvolvimento. Por isso, neste trabalho foi empregado um esforço adicional para tratar a apresentação do conteúdo ao usuário. Dentre as vantagens proporcionadas pelo CX, pode-se destacar o mecanismo de tratamento contextual, o qual tem a característica de basear-se no contexto de acesso do usuário para recomendar um conteúdo que mais seja adequado às suas necessidades. O CX utiliza mecanismos de tratamento contextual para recomendar um conteúdo personalizado ao usuário.

O CX usa uma arquitetura modular para tratamento das informações a serem apresentadas aos usuários. Os módulos no processo de acesso são implementados utilizando uma abordagem fracamente acoplada, em que módulos com interfaces bem definidas podem ser reimplementados e substituídos facilmente do sistema, se necessário.

3.3 Mecanismo de Tratamento Contextual

Uma importante característica da etapa de acesso no CX é a capacidade de basear-se no contexto de acesso para, a partir daí, exibir determinado conteúdo ao aluno. O CX utiliza cinco dimensões de contexto para identificar o acesso do usuário. São elas:

- o local em que o usuário está realizando o acesso;
- a razão ou motivo pelo qual o usuário está acessando o sistema;
- a largura de banda de rede que o usuário está utilizando;
- o tempo disponível pelo usuário para realizar o acesso; e,
- o tipo de dispositivo pelo qual o usuário está realizando o acesso.

As informações de contexto utilizadas neste trabalho, seguem a abordagem de [Morse e Dey 2000], em que é proposto utilizar cinco dimensões de contexto — *What, Who, Where, When, Why* — para representar o acesso do usuário. A Figura 3.5 mostra os valores que podem ser assumidos em cada uma dessas dimensões.

Sempre que o usuário acessa o sistema, é solicitado que ele informe o seu contexto. Na implementação atual, algumas informações são capturadas automaticamente pelo sistema, enquanto outras, o usuário precisa informar manualmente. Esse contexto é armazenado em base de dados relacional onde será consultado, posteriormente, por mecanismos que irão buscar o conteúdo que melhor se adapte a esse contexto. A Figura 3.6 ilustra o processo de tratamento contextual proposto pelo CX.

O Módulo de Tratamento Contextual (1) é responsável por tratar todas as informações referentes ao contexto do usuário. No momento em que o usuário acessa o sistema, o seu contexto é capturado e armazenado em um repositório de contextos. Essas informações são repassadas para dois módulos do sistema responsáveis por selecionar os conteúdos que

Local Universidade Casa Trabalho	Motivo Avaliação Revisão Rápida Estudo Normal	Velocidade de conexão Baixa Média Alta
Tempo disponível 15min 30min 1hr 2hr		Dispositivo Tablet Celular Desktop

Figura 3.5: Dimensões de contexto tratado no *Classroom eXperience*.

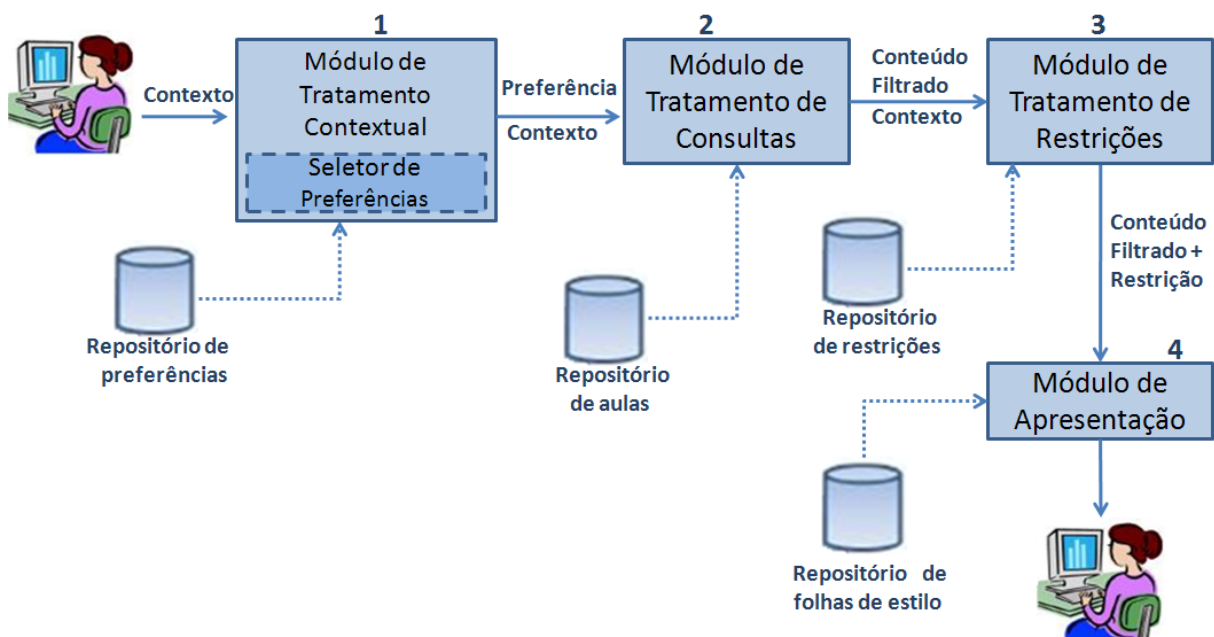


Figura 3.6: Mecanismo de Tratamento Contextual.

sejam mais adequados ao contexto do usuário: primeiro para o Módulo de Tratamento de Consultas (2) e, em seguida, para o Módulo de Tratamento de Restrições (3).

3.3.1 Módulo de Tratamento de Consultas

O Módulo de Tratamento de Consultas tem a função de selecionar um determinado conteúdo a ser apresentado ao usuário, baseado no seu contexto de acesso. Esse módulo recebe o contexto do usuário, busca o conteúdo a ser apresentado no repositório de aulas e repassa esse conteúdo para os módulos seguintes. Por ser um módulo fracamente acoplado ao sistema, sua abordagem para seleção do conteúdo pode ser alterada e substituída, dependendo da política desejada.

Neste trabalho, esse módulo foi implementado utilizando a linguagem de consulta CPref-SQL⁶. Esta é uma extensão da linguagem de consulta SQL padrão com suporte

⁶CPref-SQL (*Conditional Preference SQL*) - <http://www.lsi.ufu.br/cprefsq>

a preferências condicionais. Essa linguagem permite expressar consulta por preferências condicionais por meio de uma abordagem qualitativa. Mais detalhes sobre a linguagem de consulta CPref-SQL podem ser vistos em [Pereira e de Amo 2010].

O objetivo desse módulo é possibilitar que usuários possam informar suas preferências no sistema para que a consulta seja personalizada segundo elas. Em um determinado momento, os usuários informarão suas preferências para cada contexto de acesso, ou seja, o usuário informará o que ele prefere receber, caso acesse o sistema em determinada condição. Por exemplo, o usuário pode informar que prefere que o sistema exiba as aulas mais importantes e mais fáceis, se ele dispuser de pouco tempo disponível. Essa preferência ficará armazenada em um repositório de preferências. Posteriormente, quando o usuário acessar o sistema com esse contexto, serão recomendadas as aulas que mais se adaptem às suas preferências.

Uma outra situação, ocorre quando o usuário acessar o sistema com um contexto que não seja idêntico a algum outro contexto já cadastrado. Nesse caso, o sistema possui a capacidade de selecionar a preferência que seja mais adequada ao seu contexto atual.

Após identificada a melhor preferência, é gerada uma consulta no formato da linguagem CPref-SQL que é processada pelo Sistema de Gerenciamento de Banco de Dados e retornado um conjunto de tuplas correspondente à preferência do usuário. Assim que a consulta é processada, o conjunto de tuplas de resultados é recebido e enviado para o próximo módulo do sistema, o Módulo de Tratamento de Restrições.

A Figura 3.7 apresenta uma tela do sistema que recomenda um determinado conteúdo, baseado nas preferências do usuário. Nesse caso, o aluno informou que preferia que o sistema recomendasse às aulas mais fáceis e mais importantes às mais difíceis e menos importantes.

3.3.2 Módulo de Tratamento de Restrições

O Módulo de Tratamento de Restrições tem a função de produzir um conteúdo a ser apresentado ao usuário, baseado no seu contexto de acesso. As restrições não atuam sobre o que será apresentado, e sim, como será apresentado. A proposta é que o Módulo de Tratamento de Consultas disponibilize referências (*links*) às aulas mais relevantes ao contexto do usuário, ao passo que o Módulo de Tratamento de Restrições decidirá quais partes (*mídias*) do conteúdo capturado será disponibilizado, de acordo com o seu contexto de acesso (somente os *slides* em miniatura, *slides* + vídeo, *slides* + áudio etc).

Às vezes, o ambiente em que o aluno está realizando o acesso ao sistema implica algumas restrições, como, por exemplo restrições de velocidade de conexão, restrições do tamanho da tela do dispositivo, entre outras. Imagine que o aluno esteja fazendo o acesso de um dispositivo que tenha baixa velocidade de conexão com a Internet. Nesse caso, o fluxo de vídeo não será apresentado adequadamente, pois, nessas condições, o

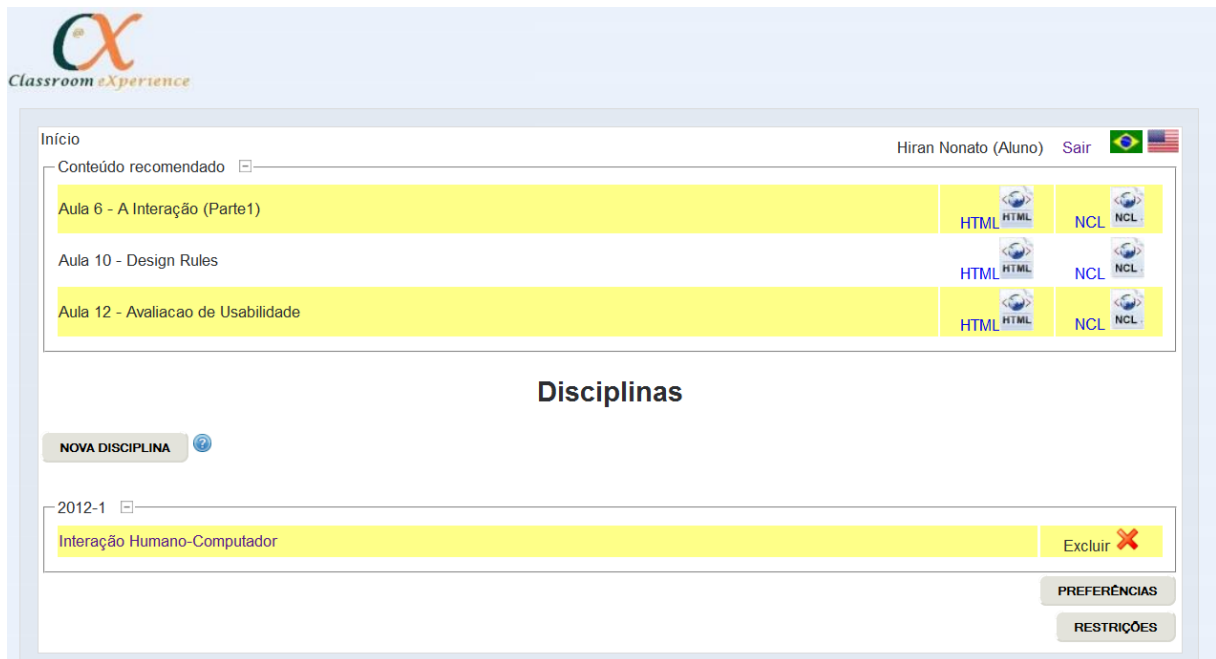


Figura 3.7: Recomendação baseada nas preferências do usuário.

sistema apresenta um conteúdo sem vídeo, somente com *slides*. Outra situação, seria, por exemplo, se o aluno estiver realizando o acesso de um dispositivo móvel (como um celular) talvez não seja possível que ele tenha acesso aos *slides* da aula, pois não conseguirá ver as informações de forma clara devido ao tamanho da tela; neste caso, o sistema poderá recomendar somente o áudio ou o vídeo da aula. Essas são algumas restrições que podem ser impostas considerando o contexto de acesso do aluno. Baseado nessas restrições, a infraestrutura aqui proposta é capaz de recomendar conteúdo ao usuário, baseado nas informações de seus contextos. O sistema pode ter algumas restrições padrões — como, apresentar somente os slides da aula caso o usuário acesse o sistema com uma velocidade de conexão com a Internet baixa; apresentar somente o áudio da aula caso o usuário acesse o sistema por meio de um celular — que são tratadas caso o usuário não informe o que deseja.

Uma vez selecionada a restrição mais adequada, essa informação é enviada para o Módulo de Apresentação, que irá gerar o conteúdo a ser disponibilizado ao usuário de acordo com a restrição selecionada.

3.3.3 Módulo de Apresentação

O Módulo de Apresentação é responsável por gerar os documentos que serão apresentados ao usuário. Esse módulo recebe o conteúdo selecionado nos módulos anteriores e produz documentos HTML que correspondam aos interesses dos usuários. Esse módulo recebe o conteúdo filtrado pelo Módulo de Consulta adicionado à restrição selecionada no Módulo de Restrição, a qual contém a descrição de qual folha de estilo será usada para a geração do documento a ser apresentado.

O sistema possui um repositório de folhas de estilo que correspondem aos resultados que podem ser apresentados aos alunos. O Módulo de Apresentação busca neste repositório a folha de estilo indicada na restrição selecionada pelo Módulo de Tratamento de Restrições e gera o documento a ser apresentado ao usuário. Esse processo é repetido para cada acesso pelos alunos no sistema.

A Figura 3.8 mostra duas exibições geradas a partir de restrições diferentes. Enquanto que a Figura 3.8(A) apresenta somente os *slides* da aula, a Figura 3.8(B), apresenta os *slides* acrescentados do vídeo.

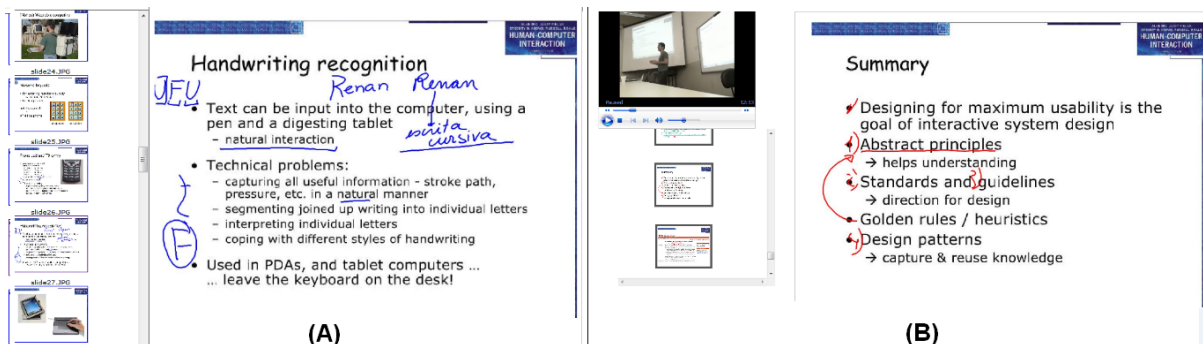


Figura 3.8: Diferentes formas de apresentação.

3.4 Visão Geral da Interação Entre as Etapas do Sistema

Como citado anteriormente, o processo de capturar e disponibilizar essas informações capturadas é estruturado em quatro fases: pré-produção, gravação ao vivo, pós-produção e acesso. Esta seção mostra como ocorre a interação entre cada uma dessas etapas.

Como observado na Figura 3.9, a Etapa 1 consiste no cadastro das informações referentes a uma aula. O professor terá acesso a uma plataforma Web que possibilita o gerenciamento de todo o conteúdo referente às suas disciplinas. Tendo já preparado o conteúdo a ser ministrado, por exemplo, *slides* Powerpoint™ ou um documento PDF, o professor pode cadastrá-lo na aplicação. A Figura 3.10 mostra a área disponível para cadastro de uma nova aula.

Essas informações cadastradas são então armazenadas em servidores disponíveis na rede (Etapa 2). A partir desse momento, o professor pode se dirigir a alguma sala instrumentada para realizar a captura de sua aula.

A próxima etapa, Etapa 3, ocorre quando o professor chega a uma sala de aula instrumentada e deseja iniciar o processo de captura. Nesse momento, o professor acessa sua plataforma onde poderá ter acesso a todas as aulas previamente cadastradas. O professor escolhe uma aula para ministrar. Nesse instante, uma aplicação cliente é enviada para o computador da sala onde se iniciará o processo de captura. Após finalizado o processo

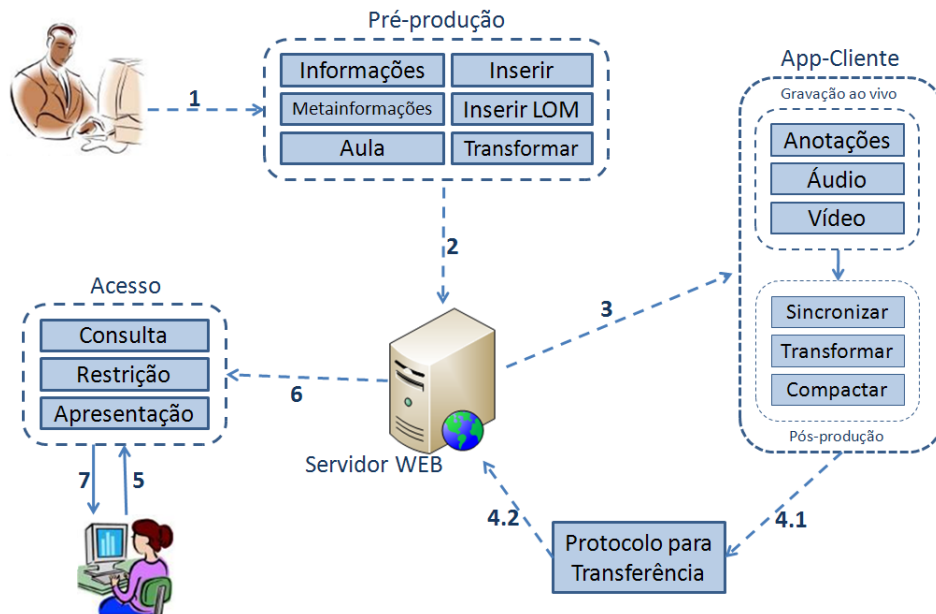


Figura 3.9: Interação entre as etapas da aplicação.

de captura, inicia-se o processo de pós-produção, em que serão sincronizadas todas as informações e gerados documentos para apresentação.

A seguir, inicia-se a Etapa 4. A aplicação cliente instancia um mecanismo para a transferência do conteúdo produzido (Etapa 4.1). Esse mecanismo utiliza um protocolo de transferência que busca de forma transparente por servidores de armazenamento disponíveis na rede (Etapa 4.2). Essa busca acontece de forma automática, sem a necessidade de informar onde está localizado o serviço de armazenamento. O protocolo de transferência foi desenvolvido no decorrer deste projeto e é mostrado em detalhes na Seção 4.

A Etapa 5 consiste no acesso das informações pelo aluno. Uma vez capturado e transferido, o conteúdo já está disponível para revisão. A Etapa 6 realiza a busca e recuperação do conteúdo e a Etapa 7 consiste na apresentação deste conteúdo ao aluno. Dessa forma, os usuários acessam uma interface Web em que estarão disponíveis todas as aulas agrupadas por disciplina. Nessa mesma interface também são feitas as recomendações de conteúdo conforme as preferências do usuário. A Figura 3.11 mostra um exemplo de uma área disponível ao aluno para acesso às aulas capturadas, neste caso, estão sendo apresentadas todas as aulas da disciplina Interação Humano-Computador do segundo semestre de 2011. A partir disso, o aluno pode selecionar o conteúdo que deseja visualizar em diferentes formatos.

Após a apresentação de todas as funcionalidades do CX, pôde-se observar que as mesmas aderem aos quatro objetivos propostos por [Truong e Hayes 2009], apresentados na Seção 2.6. O primeiro objetivo informa que a captura deve acontecer naturalmente, o CX atende a esse objetivo, pois permite que o professor ministre sua aula sem alterar a forma tradicional como fazia. O segundo objetivo refere-se ao acesso das informações capturadas, que devem estar facilmente disponíveis ao usuário, as interfaces propostas neste trabalho

Classroom eXperience

Início / Interação Humano-Computador - 2012-1 / Nova aula Renan Cattelan (Professor) Sair

Interação Humano-Computador - 2012-1

Dados básicos

Título*:

Palavras-chave*:

Arquivo de slides: Selecionar arquivo... (.pdf, .ppt, .zip)

Dados complementares

Resumo:

Língua: Portugues

Duração: 100 min.

Nível de dificuldade: Médio

Importância: Média

LOM

*Preenchimento obrigatório

Figura 3.10: Tela de cadastro de uma aula (Pré-produção).

permitem o acesso de forma fácil e organizada, a qualquer hora e em qualquer local. O terceiro objetivo relata que os registros devem ser indexados, para facilitar o processo de busca das informações produzidas, o CX também visa atender a esse requisito, pois utiliza uma interface de acesso em que as aulas são classificadas e indexadas de forma clara e fácil, distribuídas por disciplinas e semestre, além da recomendação de conteúdo baseado no contexto de acesso do aluno. O quarto e último objetivo sugere que os registros devem ser produzidos de modo ético e legal, o sistema proposto aqui visa atender esse objetivo, ao permitir que o conteúdo capturado seja apresentado somente a partir do consentimento do professor, diferenciando assim, o conteúdo público do privado. Também é feito controle sobre quem acessa o conteúdo, com a disponibilização, pelo professor, de uma senha de acesso ao conteúdo de cada disciplina.



Classroom eXperience

Início / Interação Humano-Computador - 2012-1 Hiran Nonato (Aluno) Sair

Interação Humano-Computador - 2012-1

Título: Pesquisar

Título	Data da captura	Acesso		
Aula 1 - Apresentação da disciplina	09/05/2012	HTML Básico	HTML Personalizado	NCL
Aula 2 - O Humano (parte1)	09/05/2012	HTML Básico	HTML Personalizado	NCL
Aula 3 - O Humano (parte2)	29/04/2012	HTML Básico	HTML Personalizado	NCL
Aula 4 - O Computador (parte1)	29/04/2012	HTML Básico	HTML Personalizado	NCL
Aula 5 - O Computador (parte2)	29/04/2012	HTML Básico	HTML Personalizado	NCL
Aula 6 - A Interação (Parte1)	29/04/2012	HTML Básico	HTML Personalizado	NCL
Aula 10 - Design Rules	02/05/2012	HTML Básico	HTML Personalizado	NCL

Figura 3.11: Área do aluno para acesso às aulas capturadas.

3.5 Considerações Finais

Este capítulo apresentou o *CX*, sistema desenvolvido neste trabalho para apoiar atividades de C&A em ambientes educacionais. Foi detalhada sua arquitetura, pontuando o funcionamento de cada um dos seus módulos. Foi detalhado todo o processo de estruturação dos módulos, bem como, algumas questões de implementação.

O sistema é composto por recursos que o tornam bastante flexível, pois cada módulo do sistema é implementado de forma fracamente acoplada, sendo possível sua reimplementação dependendo da abordagem a ser utilizada. O capítulo mostra também o funcionamento do mecanismo de tratamento de contexto, em que o Módulo de Tratamento Contextual auxilia os Módulos de Tratamento de Consultas e Restrições na recomendação de conteúdo ao usuário. Por fim, foram apresentadas as interações entre cada etapa no processo de capturar, gerar documentos derivados, armazenar e acessar o conteúdo capturado, em que foi possível observar o papel e a importância de cada fase neste processo.

Capítulo 4

Modelo de Comunicação Utilizado no *Classroom eXperience*

4.1 Considerações Iniciais

Um dos grandes desafios no desenvolvimento de aplicações de captura e acesso (C&A) está na capacidade de prover aos usuários alta disponibilidade do conteúdo, acessar as informações a qualquer hora e em qualquer lugar, com a certeza de que estarão sempre disponíveis. Isso pode ser alcançado por meio de uma estruturação na forma de comunicação entre os componentes que produzirão e armazenarão o conteúdo capturado.

Existem diversas abordagens para a estruturação da comunicação em sistemas de C&A. Alguns trabalhos utilizam arquitetura cliente/servidor (C/S), pela qual os clientes capturam o conteúdo e o transmitem para servidores responsáveis pelo seu armazenamento [Nishiuchi et al. 2010, Kunz et al. 2010, Giliberti et al. 2011]. Outras formas de comunicação podem ser vistas em [Kameas et al. 2002, Xie et al. 2010, Barolli e Xhafa 2011], em que a comunicação é baseada no paradigma de computação *peer-to-peer* (P2P). Nesses casos, geralmente, um computador pode atuar simultaneamente como cliente e servidor. Em ambos os casos, o principal objetivo é fazer com que o conteúdo esteja disponível ao usuário sempre que necessário.

Neste capítulo é apresentado o mecanismo de comunicação que foi desenvolvido para o *Classroom eXperience* (CX). Trata-se de um protocolo para a transferência e a sincronização de conteúdo que usa uma abordagem P2P. Na Seção 4.2 é mostrada a arquitetura utilizada na infraestrutura proposta; na Seção 4.3 é exposta toda a especificação do modelo de comunicação desenvolvido; os detalhes de implementação são discutidos na Seção 4.4; e, finalmente, na Seção 4.5 são apresentadas as considerações finais sobre o capítulo.

4.2 Arquitetura

Nas versões iniciais do CX, objetivava-se criar um mecanismo de comunicação integrado à plataforma de captura e um serviço de armazenamento restrito ao cenário aqui proposto. Com o desenvolver do trabalho, houve a percepção de que seria possível a criação de um arcabouço mais genérico, que poderia ser aplicado em outros cenários. A partir disso, surgiu a proposta da especificação de um protocolo que pudesse ser empregado para a transferência e a sincronização de arquivos, em que a aplicação P2P de captura e armazenamento de informações utilizaria seus serviços com o menor esforço possível e de forma transparente.

O protocolo aqui proposto não se limita ao emprego apenas em ambientes de C&A, ele pode ser aplicado em diversos domínios em que são necessárias transferências de arquivos e descoberta de serviços na rede. Esse mecanismo de comunicação foi construído para atuar como uma camada sobre a plataforma JXTA [Sun Microsystems, Inc. 2007], ferramenta aberta que provê recursos para configuração e uso de redes P2P. A escolha dessa plataforma se deu principalmente pela sua estabilidade e por sua grande utilização pela comunidade acadêmica [Xhafa et al. 2008, Zhong-xian et al. 2010, Barolli e Xhafa 2011].

A plataforma JXTA¹ é composta por um conjunto de seis protocolos que são independentes de linguagem de programação, sistema operacional e topologia de rede. Os protocolos usados no JXTA são: *Peer Discovery Protocol*, *Peer Information Protocol*, *Peer Resolver Protocol*, *Pipe Binding Protocol*, *Endpoint Routing Protocol*, *Rendezvous Protocol*. Os *peers* presentes na rede podem ser organizados de forma descentralizada. Usando esses protocolos, é possível que *peers* em uma rede privada possam se conectar a *peers* na Internet, passando por barreiras de *firewalls* e NATs (*Network Address Translation*) [Barolli e Xhafa 2011].

Em JXTA, *peers* podem ser diversos dispositivos de rede, como: sensores, celulares, *Tablet PCs*, PCs, entre outros. Esses *peers* podem ser organizados em grupos, ou seja, um conjunto de *peers* que fazem o uso de serviços comuns. Cada *peer* que se associa a um determinado grupo deve implementar todos os serviços do grupo. Os *peers* se auto-organizam nesses grupos, nos quais cada grupo é identificado por um ID e pode estabelecer sua própria política de membros.

JXTA define um conjunto de serviços básicos, entre os quais o serviço de descoberta (*discovery*), utilizado pelos membros dos grupos para a busca de recursos, tais como: serviços, *pipes*, *peers* e grupos. Outro importante serviço disponível é o *pipe service*, que é usado para criar e gerenciar conexões de *pipes* entre *peers* membros de um grupo. *Pipes* são mecanismos de transferência de mensagem síncronos e unidirecionais para comunicação entre *peers*. *Pipes* são tratados como canais de comunicação virtual e podem conectar

¹<http://java.net/projects/jxta>

peers que não tenham uma conexão física direta, criando assim, uma conexão lógica. Eles podem ser empregados para a transferência de qualquer tipo de dados, incluindo textos HTML, imagens, músicas, código binário, *strings* e objetos de dados.

Todos os recursos na rede JXTA são representados por *advertisements*: estruturas de metadados neutras descritas como documentos XML. Os *peers* fazem o *cache*, a publicação e a troca de *advertisements* para descobrir e localizar recursos disponíveis na rede [Cattelan 2009]. A Figura 4.1 mostra um *advertisement* utilizado para representar um *pipe*.

```
1 <?xml version="1.0"?>
2 <!DOCTYPE jxta:PipeAdvertisement>
3
4 <jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
5   <Id>
6     urn:jxta:uuid-59616261646162614E50472050325033D4AEE339F...
7   </Id>
8   <Type>
9     JxtaUnicast
10  </Type>
11  <Name>
12    PipeTest
13  </Name>
14 </jxta:PipeAdvertisement>
15
```

Figura 4.1: *Advertisement* para representação de um *pipe* na rede.

A arquitetura proposta para o protocolo desenvolvido permite a utilização de um ou mais serviços de armazenamento (consumidores), bem como um ou mais ambientes de captura (produtores). Os *peers* produtores buscam por *peers* consumidores para armazenar o conteúdo produzido durante uma sessão de captura, da mesma forma que os *peers* consumidores buscam por *peers* também consumidores para realizar a sincronização do conteúdo já armazenado. Uma característica importante da arquitetura desse modelo de comunicação é a capacidade de comunicação entre *peers* em redes distintas, mesmo que estejam separados fisicamente ou logicamente através de *firewalls* e NATs.

Como dito anteriormente, o mecanismo de comunicação proposto neste trabalho é composto de um protocolo que atua como uma camada, entre o JXTA e a aplicação P2P que manipula a transferência de conteúdo. Essa camada foi denominada de CAL (*Content Abstraction Layer*). CAL usa recursos do JXTA para prover transferência e sincronização de conteúdo de forma simples para a aplicação P2P de captura, propiciando o armazenamento do conteúdo capturado. Os detalhes referentes ao protocolo serão descritos nas próximas seções.

4.3 Especificação do Protocolo

[Holzmann 1991] define cinco elementos que são considerados essenciais em um protocolo para o contexto de redes de computadores. O primeiro elemento é o ambiente, que leva em consideração as características físicas e lógicas onde o protocolo atuará, sua arquitetura, organização etc. O segundo elemento é o vocabulário, um conjunto de eventos (nome das mensagens) usados para especificar as transições de estado do protocolo. O terceiro elemento é o formato, que mostra a estrutura de cada mensagem a ser transmitida. O quarto elemento são os serviços, elementos abstratos que definem as funcionalidades do protocolo. Por fim, regras procedimentais, que definem o comportamento do protocolo por meio de especificações formais, por exemplo, autômatos.

A seguir, será descrita a forma como cada elemento foi projetado no protocolo aqui proposto.

4.3.1 Ambiente

O protocolo aqui apresentado foi projetado para ser usado sobre uma rede P2P em que cada *peer* pode atuar como produtor e/ou consumidor de informações. O produtor é o *peer* que captura conteúdo multimídia do mundo real por meio de dispositivos instalados no ambiente e produz artefatos de dados referentes a essas sessões de captura.

O consumidor é o *peer* que armazena as informações recebidas do produtor e também as compartilha com outros consumidores. Assim, o *peer* consumidor pode compartilhar seu conteúdo com outros *peers* para prover redundância e o *peer* produtor pode enviar seu conteúdo para torná-lo disponível a outros *peers* na rede. A Figura 4.2 mostra um exemplo de disposição do ambiente com alguns *peers* consumidores e produtores.

No exemplo da Figura 4.2, os *peers* P1, P2 e P3 atuam na rede como *peers* produtores, ou seja, estão no ambiente instrumentado para realizar a captura das informações. O *peer* P4 também é produtor, mas além disso, ele pode ser um *peer* consumidor, provendo serviço de armazenamento. Os *peers* P5 e P6 são consumidores; eles atuam somente como serviços de armazenamento do conteúdo produzido pelos *peers* produtores.

4.3.2 Vocabulário

Depois dessas considerações sobre o ambiente, é formalizado o vocabulário do protocolo aqui proposto, que define a semântica de mensagens usadas na comunicação [Holzmann 1991].

O conjunto de mensagens do protocolo consiste em:

- *start_session_req*: mensagem enviada pela aplicação P2P para solicitar a criação de uma sessão entre dois *peers*. Essa sessão é necessária para realização de qualquer operação entre os *peers* participantes.

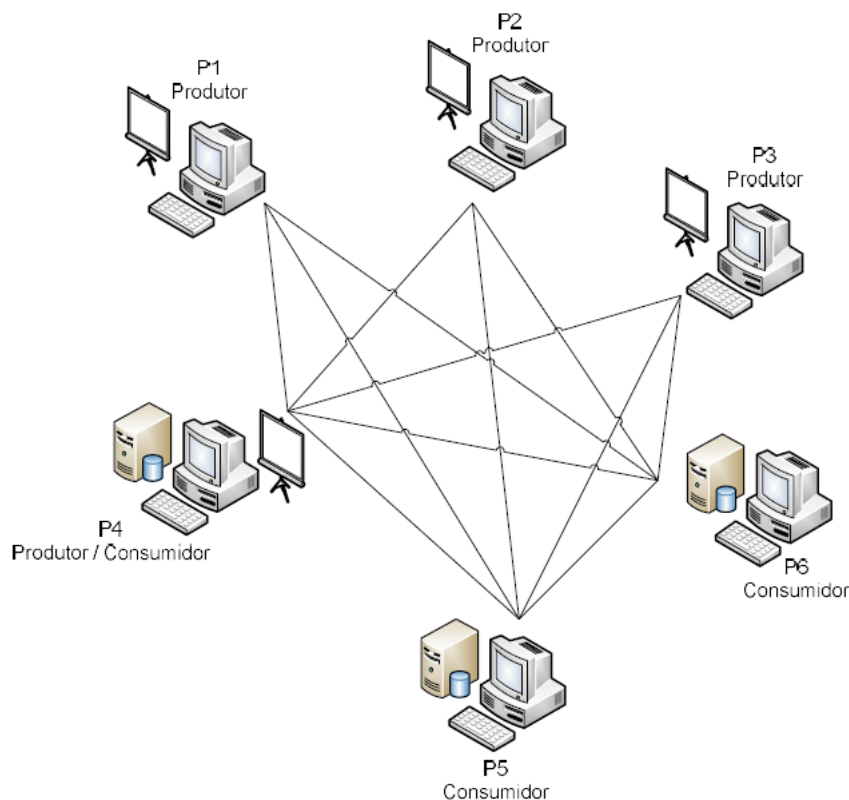


Figura 4.2: Ambiente com *peers* produtores e consumidores.

- *connect_req*: mensagem enviada pela camada, com o propósito de solicitar uma conexão com outro *peer* na rede.
- *discovery_req*: mensagem enviada pela camada para procurar por *peers* disponíveis na rede.
- *end_session_req*: mensagem enviada pela aplicação P2P para encerrar uma sessão estabelecida entre dois *peers*. O protocolo recebe essa mensagem e envia um *disconnect_req* para concluir a operação.
- *disconnect_req*: mensagem usada para encerrar uma conexão entre dois *peers*.
- *ls_request*: mensagem enviada para consultar a lista de conteúdo (lista de arquivos) de outro *peer*. É enviada pelas aplicações P2P que atuam como *peers* consumidores. Essa é a primeira etapa para dois *peers* sincronizarem seus conteúdos;
- *send_seg_request*: mensagem usada para enviar um segmento de um determinado conteúdo. Quando o conteúdo a ser enviado é muito grande, então ele é dividido em pequenas partes antes de ser enviado. Esse serviço pode ser usado tanto pelo produtor quanto pelo consumidor;
- *send_seg_response-*: essa mensagem representa um *send_seg_request* não reconhecido. Ela é enviada quando o último *send_seg_request* não for reconhecido.
- *send_seg_response+*: essa mensagem representa um *send_seg_request* reconhe-

cido.

- *send_msg_req*: mensagem utilizada para requisitar algum conteúdo por meio de uma mensagem.
- *ft_get_req*: mensagem usada para requisitar um conteúdo específico em outro *peer*. Essa mensagem pode ser enviada somente por um *peer* consumidor;
- *ft_put_req*: mensagem empregada para enviar algum conteúdo para outro *peer* na rede.
- *ls_conf+*: mensagem usada para indicar que o último *ls_request* foi reconhecido.
- *ls_conf-*: mensagem utilizada para indicar que o último *ls_request* não foi reconhecido.
- *ft_get_conf+*: mensagem usada para indicar que o último *ft_get_request* foi reconhecido.
- *ft_get_conf-*: mensagem empregada para indicar que o último *ft_get_request* não foi reconhecido.
- *ft_put_conf+*: mensagem utilizada para indicar que o último *ft_put_request* foi reconhecido.
- *ft_put_conf-*: mensagem usada para indicar que o último *ft_put_request* não foi reconhecido.
- *send_seg_conf+*: mensagem empregada para indicar que o último *send_seg_request* foi reconhecido.
- *send_seg_conf-*: mensagem utilizada para indicar que o último *send_seg_request* não foi reconhecido.

4.3.3 Formato

Todas as mensagens do vocabulário são codificadas por um método orientado a caractere e tem o formato mostrado na Figura 4.3.

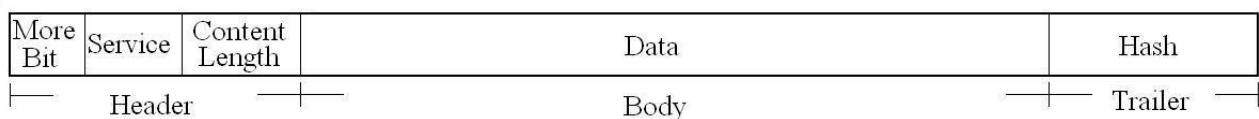


Figura 4.3: Primitiva para o protocolo proposto.

Uma primitiva neste protocolo é dividida em três partes: *header*, *body* e *trailer*. A primeira parte, o *header*, é composta por três campos: o campo *more_bit* indica se existem mais segmentos que completem a informação encaminhada; o campo *service* representa o

nome do serviço chamado, o qual é codificado em um número binário. Como o protocolo possui oito serviços disponíveis, então 3 *bits* são necessários para representar todos os serviços ($2^3 = 8$); por fim, no *header*, há o campo *content length*, que apresenta o tamanho total do próximo campo, o campo *data*.

O campo *data* contém os dados em questão a serem transferidos. Esse campo pode assumir o tamanho de até 64 *Kbytes*.

Por fim, encontra-se o campo *trailer*, que armazena um código *hash* para checagem de erro. Por utilizar-se de uma criptografia MD5, são necessários 16 *bytes* para esse campo.

4.3.4 Serviços e Comportamentos

Agora serão definidos os oito serviços providos por este protocolo: *publish*, *start_session*, *end_session*, *list_status*, *ft_get*, *ft_put*, *send_seg* e *reject*.

O primeiro serviço, *publish*, é usado para disponibilizar um *peer* consumidor na rede. Isso significa que um novo *peer* deseja prover seu espaço em disco para armazenar conteúdo para outro *peer*. Esse serviço não é confirmado, ou seja, o *peer* envia essa primitiva e não aguarda por confirmação.

Um dos grandes potenciais deste protocolo está na possibilidade de descoberta automática de *peers* na rede, sem a necessidade de conhecer o seu endereço (IP, ID). O serviço *start_session* representa uma sessão de comunicação estabelecida entre dois *peers*. Qualquer *peer* que deseje realizar comunicação com outro *peer* deve utilizar esse serviço. A Figura 4.4 mostra o seu autômato.

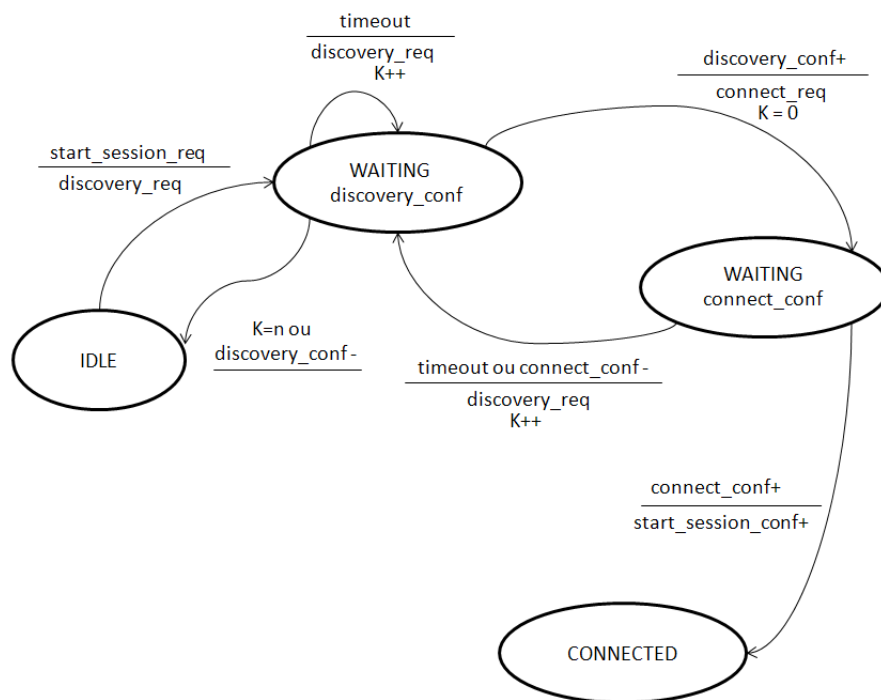


Figura 4.4: Autômato do serviço *start_session* [Araújo et al. 2012].

Estando a camada CAL no estado *IDLE*, a aplicação P2P pode enviar uma requisição de *start_session*, utilizando a mensagem *start_session_req* para criar uma nova sessão com outro *peer*. Nesse momento, a CAL envia um *discovery_req* e fica no estado *WAITING DISCOVERY_CONF*. Nesse estado, a camada está procurando por *peers* disponíveis na rede. Caso o *discovery_conf+* não aconteça em um tempo predeterminado, um novo *discovery_req* é enviado, e uma variável K (número de *timeouts*) é incrementada. Caso a variável K se iguale à variável N (número máximo de vezes em que pode ocorrer o *timeout*) ou o *discovery_conf-* é recebido, a camada volta ao estado *IDLE*, isso denota que não foram encontrados *peers* disponíveis na rede. Caso contrário, *discovery_conf+* é recebido, ou seja, um *peer* foi encontrado, então, a camada envia um *connect_req* para tentar estabelecer uma conexão com este *peer* e fica em um estado *WAITING CONNECT_CONF*. Se um *timeout* ou *connect_conf-* for recebido, a camada volta para o estado *WAITING DISCOVERY_CONF*, que tem a função de procurar por outro *peer*. Caso o *connect_conf+* seja recebido, ou seja, a conexão foi estabelecida corretamente, a camada passa para o estado *CONNECTED*. Nesse momento, é retornada uma confirmação *start_session_conf+* para a aplicação P2P, indicando sucesso na operação.

Quando uma aplicação deseja encerrar a conexão com outro *peer*, o serviço *end_session* é chamado. A camada recebe a mensagem *end_session_req* da aplicação P2P e envia um *disconnect_req* para o outro *peer* na rede, passando assim do estado *CONNECTED* para o estado *IDLE*. O comportamento desse serviço pode ser observado a partir do autômato mostrado na Figura 4.5. Nota-se que este não é um serviço confirmado.

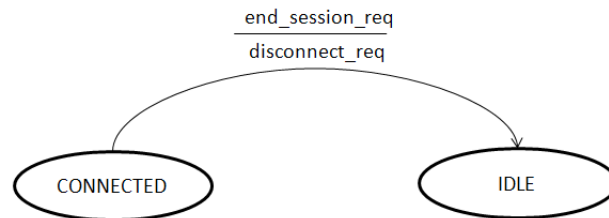


Figura 4.5: Autômato do serviço *end_session* [Araújo et al. 2012].

O serviço *list_status* é responsável por consultar o conteúdo disponível em outro *peer*. O serviço retorna para a aplicação uma lista com todo o conteúdo disponível no *peer* consultado. Esse é um serviço disponível somente para os *peers* consumidores. O autômato desse serviço é mostrado na Figura 4.6.

Inicialmente, estando a CAL no estado *CONNECTED*, recebe um *ls_req* da aplicação P2P e envia um *send_msg_req*, passando para o estado *WAITING LIST*. Nesse estado, ela pode receber um aviso de *timeout* ou um *ls_conf-*, o qual retornará para o estado *CONNECTED* e avisará à aplicação P2P que ocorreu um erro na requisição. Caso não ocorra erro nessa fase do processo, um *ls_conf+* será recebido, indicando que a requisição foi recebida pelo outro *peer*. Nesse momento, a CAL envia um *send_msg_req* com o con-

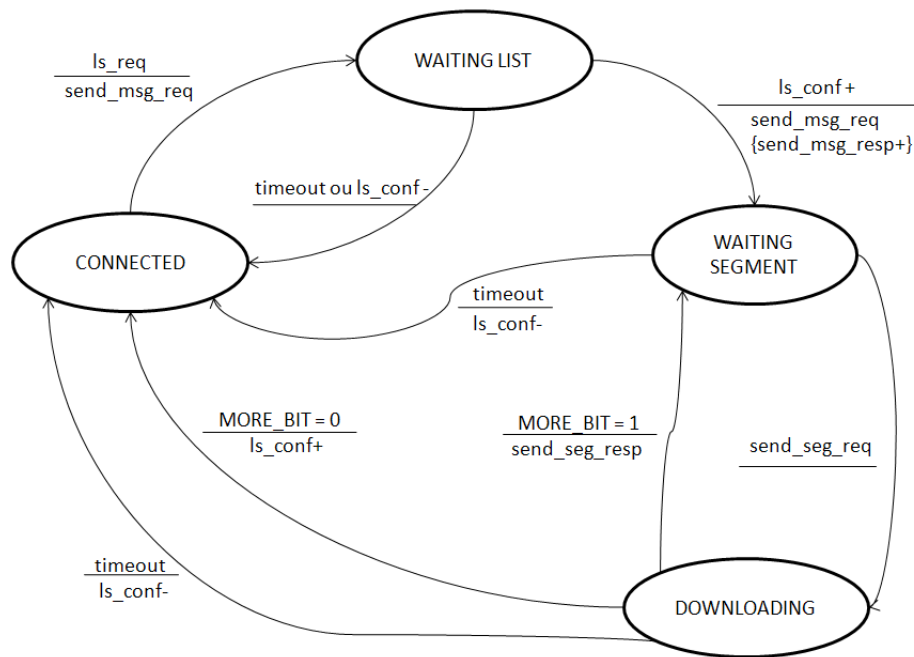


Figura 4.6: Autômato do serviço *list_status* (adaptada de [Araújo et al. 2012]).

teúdo “send_msg_resp+” dentro do campo *data* para avisar que recebeu a confirmação e passa para o estado *WAITING SEGMENT*, o qual representa que ele está pronto para receber os segmentos de informação. Estando neste estado, caso a camada receba um aviso de *timeout*, ela enviará um *ls_conf-* para a aplicação, informando que houve erro e passará para o estado *CONNECTED*. Caso não ocorra *timeout*, um *send_seg_req* será recebido com um segmento da informação dentro da mensagem, e a camada passará para o estado *DOWNLOADING*, no qual fará todo o processo de *download* desse segmento. Se houver mais pacotes a serem recebidos (campo *more_bit* = 1), a camada enviará um *send_seg_resp*, informando ao outro *peer* que já está disponível para receber o próximo pacote. Esses passos se repetem até que o campo *more_bit* seja igual a 0 (zero), nesse caso, será enviado um *ls_conf+* para a aplicação, informando que houve sucesso na transferência. Caso um *timeout* ocorra, será enviado um *ls_conf-* para a aplicação, informando que houve falha.

Para melhor ilustrar, a Figura 4.7 mostra um diagrama de ordem temporal para o serviço *list_status*. É importante ressaltar que as mensagens que chegam na camada são transmitidas dentro do campo *data* da primitiva *send_msg* do JXTA. Esse comportamento pode também ser observado em outros serviços, uma vez que eles foram projetados para ter abstrações providas pelo JXTA.

Outro serviço disponibilizado por este protocolo é o *ft_get*, que é responsável por fazer o *download* de um determinado conteúdo disponível em outro *peer*. Como no serviço *list_status*, o *ft_get* está disponível apenas para os *peers* consumidores. A Figura 4.8 mostra o autômato que ilustra o comportamento desse serviço.

Estando a camada CAL no estado *CONNECTED*, a aplicação envia um *ft_get_req*

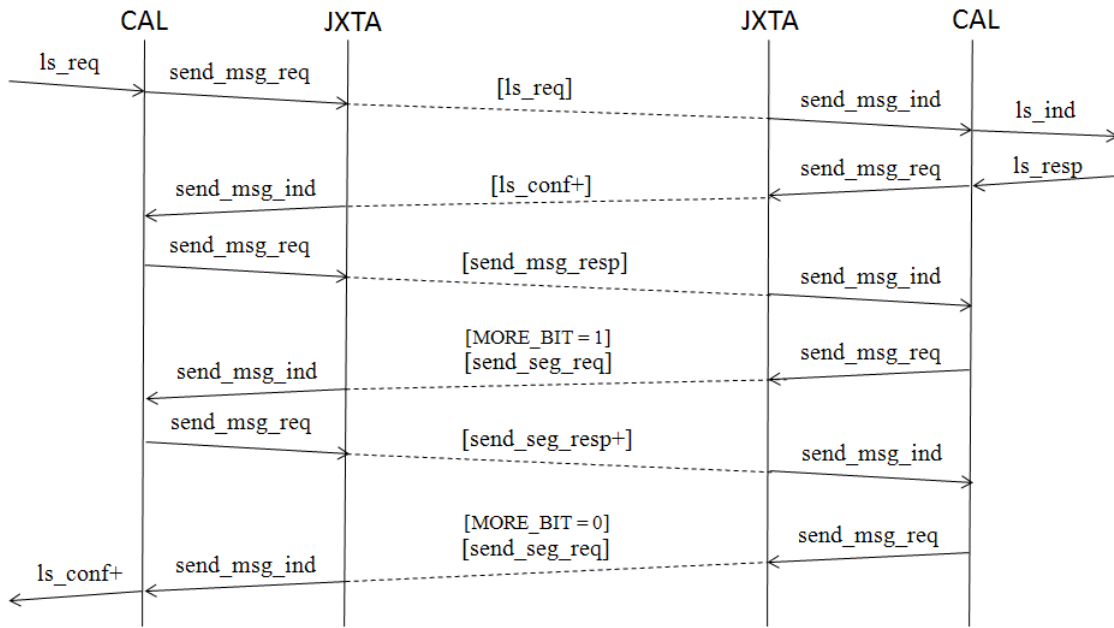


Figura 4.7: Diagrama temporal para o serviço *list_status* (adaptada de [Araújo et al. 2012]).

para requisitar o conteúdo do outro *peer* na rede. Nesse momento, a camada envia um *send_msg_request* e passa para o estado *WAITING_FT_GET_CONF* até receber a confirmação de que recebeu a requisição. Caso aconteça um *timeout* ou um *ft_get_conf-* a camada volta ao estado *CONNECTED*, abortando a requisição e envia uma mensagem para a aplicação, indicando que ocorreu uma falha. Caso um *ft_get_conf+* seja recebido, a camada envia uma mensagem de sucesso e fica no estado *WAITING_SEGMENT* aguardando pelos segmentos do conteúdo solicitado. Estando nesse estado, caso aconteça um *timeout* ou um *ft_get_conf-*, a camada aborta a requisição e retorna ao estado *CONNECTED*. Caso não ocorra *timeout*, um *send_seg_req* será recebido com um segmento da informação dentro da mensagem, e a camada passará para o estado *DOWNLOADING* no qual fará todo o processo de *download* desse segmento. Caso haja mais segmentos a serem recebidos, a camada informa ao outro *peer* que recebeu o conteúdo corretamente, e que está pronto para receber os próximos pacotes. Caso não haja mais segmentos a serem recebidos, o protocolo envia um *ft_get_conf+* para a aplicação P2P, informando que a transferência foi concluída. Caso algum erro aconteça nesse estado, a camada envia um *ft_get_conf-* para a aplicação abortando a operação. Esse processo de *download* do segmento acontece semelhante ao *list_status*.

O serviço *ft_put* está disponível somente para os *peers* produtores. O serviço é usado para enviar conteúdos a serem armazenados em outros *peers* (também devem ser consumidores). Um exemplo de uso deste serviço pode ser visto quando uma sessão de captura é finalizada, então ele é instanciado para fazer o *upload* das informações capturadas para os serviços de armazenamento. A Figura 4.9 mostra o autômato desse serviço.

No momento em que a aplicação deseja realizar uma transferência, é enviado um

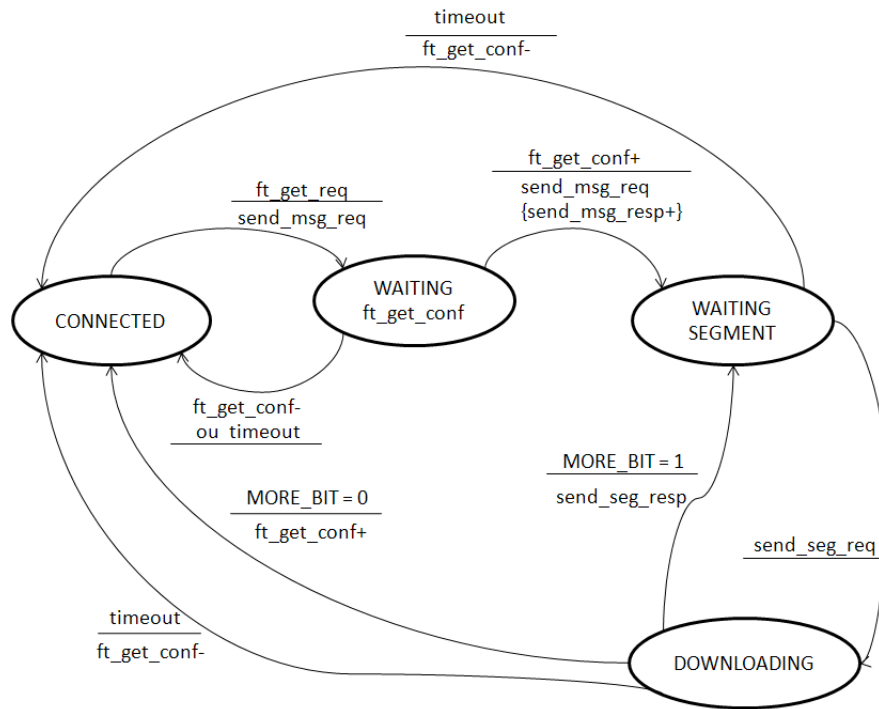


Figura 4.8: Autômato para o serviço *ft_get* (adaptada de [Araújo et al. 2012]).

ft_put_req à camada, que envia um *send_msg_req*, passando para o estado *WAITING FT_PUT_CONF*. Caso um *ft_put_conf-* ou um *timeout* é recebido, a camada retorna para o estado *CONNECTED* e informa à aplicação que a transferência falhou. Caso seja recebido um *ft_put_conf+*, a camada passa para o estado *SENDING SEGMENT* e envia um *send_seg_req*, contendo um segmento do arquivo requisitado, passando para o estado *UPLOADING*. Caso o arquivo contenha mais segmentos, e a confirmação de recebimento pelo outro *peer* chegou, então, a camada continua enviando os seus segmentos um a um até o último pacote, nesse caso, finaliza a operação enviando um *ft_put_conf+* para a aplicação, informando sucesso no processo. Se algum erro acontecer nesse processo de *upload*, a camada envia um *ft_put_conf-*, informando que houve erro na transferência passando para o estado *CONNECTED*.

O serviço, *send_seg* está disponível para os produtores e consumidores. Ele é responsável por enviar segmentos de conteúdo para outro *peer* na rede. Nesse cenário de compartilhamento de conteúdo multimídia, os dados, geralmente, são muito grandes para serem enviados em uma simples transferência. Então, dados devem ser segmentados e enviados partes por partes. Esse comportamento pode ser notado nos autômatos apresentados anteriormente nos serviços *list_status*, *ft_get* e *ft_put*.

O último serviço a ser descrito é o *reject*, o qual está disponível tanto para *peers* produtores quanto para *peers* consumidores. Ele é usado para desconsiderar primitivas que são recebidas com erros ou primitivas não reconhecidas pela CAL. Em todos os serviços mencionados acima, a camada pode receber primitivas com erros enviados pela aplicação, então, se isso ocorrer, a camada envia uma mensagem com o serviço *reject*, retorna para o

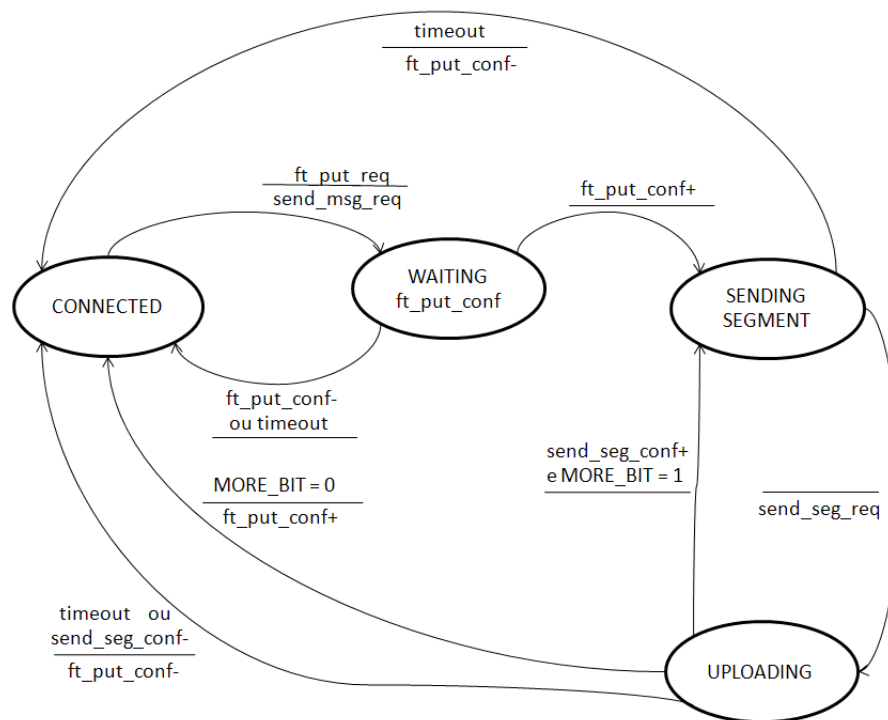


Figura 4.9: Autômato para o serviço *ft_put* (adaptada de [Araújo et al. 2012]).

mesmo estado que se encontrava e aguarda por outra mensagem. Por meio desse serviço, é possível garantir a consistência da comunicação entre os *peers*.

O desenvolvimento de um protocolo requer grande atenção na especificação de sua estrutura. Nesta seção foram apresentados os elementos principais presentes em qualquer protocolo de redes. Na próxima seção serão descritas a estrutura e algumas decisões tomadas na etapa de implementação desse modelo de comunicação.

4.4 Detalhes de Implementação

O protótipo ora apresentado foi escrito na linguagem Java. Uma das principais razões para sua escolha se deu por sua portabilidade, robustez, segurança e seu grande uso pela comunidade de desenvolvedores. Nesta seção será descrita a implementação do protocolo proposto e serão discutidas também várias decisões que foram tomadas durante o projeto e desenvolvimento deste.

Como mencionado anteriormente, o protocolo atua como uma camada localizada entre a plataforma JXTA e a aplicação P2P (Figura 4.10). A implementação foi proposta dessa forma a fim de dar total autonomia para a aplicação P2P criar sua própria política de transferência de conteúdo. Desse modo, as regras para transferências e replicação de conteúdo não são definidas pelo protocolo, mas pela aplicação que faz o uso deste protocolo. Assim, é possível, por exemplo, que uma determinada aplicação tenha uma política de sincronização diária, todo dia às 12h realiza a sincronização do conteúdo;

outra aplicação pode ter uma política de transferência diferente, com mais frequência, por exemplo, sincronizar seu conteúdo a cada 4 horas, diariamente. Portanto, o protocolo desenvolvido não impõe uma regra para sincronização, e sim, deixa flexível para cada aplicação estabelecer a que seja mais bem adequada a seu cenário.

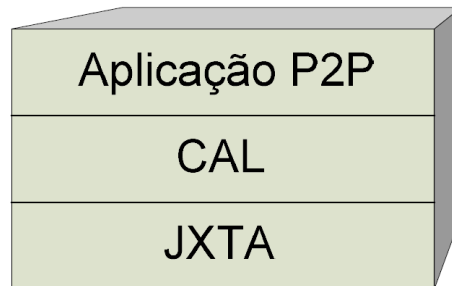


Figura 4.10: Disposição das camadas.

Cada serviço no protocolo — *publish*, *start_session*, *end_session*, *list_status*, *ft_get*, *ft_put*, *send_seg* e *reject* — foi implementado por meio de interfaces bem definidas, permitindo que as aplicações usassem suas funcionalidades com o menor esforço possível.

Para o serviço *publish* foi desenvolvida uma interface a qual é chamada para criar um serviço de armazenamento na rede. Quando esse serviço é invocado, o *peer* que o invocou fica aguardando por conexões para receber conteúdo. Essa interface usa uma primitiva nativa do JXTA para publicar um *advertisement* na rede. Um método *publish()* é utilizado para receber o nome do serviço a ser criado e chamar rotinas do JXTA para disponibilizá-lo na rede.

No serviço *start_session*, a interface é usada para estabelecer uma conexão com um serviço de armazenamento disponível na rede. Entre os vários serviços disponibilizados pelo JXTA, essa interface usa o *discovery* e o *connect*. Quando essa interface é chamada pela camada superior, o serviço *discovery* do JXTA busca por um *peer* que tem a função de serviço de armazenamento na rede e, depois, o serviço *connect* é chamado para estabelecer uma conexão entre os dois *peers*.

Um dos grandes potenciais desse protocolo é alcançado por meio do serviço *start_session*, pois a consulta do *peer* de armazenamento acontece completamente transparente para a aplicação, ou seja, a aplicação P2P não precisa chamar nenhum método além do *start_session* para se conectar a outro *peer*. A aplicação P2P pode ou não informar com qual serviço ele quer estabelecer uma conexão. Se ela não informar o nome do serviço com que deseja conectar, a conexão acontece com o primeiro serviço encontrado. Se a conexão for estabelecida com sucesso, um *JxtaBiDiPipe*² é criado entre os dois *peers*.

A interface *end_session* é responsável por finalizar uma conexão com um serviço de armazenamento. O método *end_session()* encerra a conexão do *pipe* criado no *start_session()* e libera o *peer* para criar novas conexões ou simplesmente deixar de existir.

²*JxtaBiDiPipe* são mecanismos disponibilizados pela plataforma JXTA para prover transferência de dados de forma bidirecional e confiável.

A interface *list_status* é utilizada por um *peer* consumidor para saber o conteúdo de um serviço de armazenamento (outro *peer* consumidor). A comunicação entre os *peers* acontece por meio de serviços de mensagens providos pelo JXTA, o qual é implementado pelo método *StringMessageElement()*. Os principais métodos usados por esse serviço são: *getMessageElement()*, *addMessageElement()* e *sendMessage()*. Nessa interface os dados são trafegados mediante segmentos de mensagens. O campo *more_bit* é usado para indicar se existe ou não mais conteúdo a ser recebido. Depois que todos os segmentos são recebidos, a informação é organizada e enviada para a aplicação que a requisitou.

No serviço *ft_get*, a interface é chamada por um *peer* consumidor que deseja algum conteúdo que está indisponível no seu repositório. Essa atividade pode ser executada entre dois *peers* consumidores para sincronizar o seu conteúdo. O conteúdo é enviado por meio de segmentos na rede; assim, os mesmos métodos empregados no *list_status* são usados pelo *ft_get*. Entre os diversos métodos usados nessa interface, podemos citar o *check_free_space()*, que é usado para verificar se existe espaço suficiente para receber o arquivo.

A interface do *ft_put* é usada por *peers* produtores para enviar o conteúdo para serviços de armazenamento. Como no *ls_status* e *ft_get*, os conteúdos são enviados em segmentos na rede, usando sempre o campo *more_bit* para identificar o último segmento. Nessa interface, também são usados os métodos *getMessageElement()*, *addMessageElement()* e *sendMessage()* para a transferência do conteúdo.

O conteúdo segmentado é enviado pela rede através da chamada do serviço *send_seg*. Como mencionado, esse é usado por vários serviços deste protocolo, como: *list_status*, *ft_get* e *ft_put*.

A interface do serviço *reject* é chamada sempre que um novo pacote é recebido. Ela é usada para informar quando o pacote foi recebido com erro e assim, foi rejeitado. Essa interface, bem como algumas mencionadas anteriormente, também recorre aos serviços providos pelo JXTA para enviar mensagens dentro do campo *data*.

4.5 Considerações Finais

Neste capítulo foi apresentada a estrutura do modelo de comunicação empregado para transferência e sincronização de dados no CX. Este modelo de comunicação foi estruturado a partir de um protocolo que atua como uma camada entre a plataforma JXTA e a aplicação P2P. Os serviços disponibilizados por este protocolo usam funcionalidades do JXTA para disponibilizar, transparentemente, recursos de transferência e sincronização de conteúdo com outras aplicações disponíveis na rede.

Foram discutidos conceitos sobre a plataforma JXTA, bem como, os elementos essenciais que compõem um protocolo de redes de computadores. Por fim, foram mostrados os detalhes de implementação que foram tomados durante o desenvolvimento do protocolo.

Este modelo de comunicação permite que aplicações P2P realizem transferência de dados pela rede de forma transparente. O protocolo foi estruturado de maneira genérica, e pode ser aplicado a diversos domínios, não somente a aplicações de C&A. Outra importante característica do protocolo descrito é a possibilidade da aplicação P2P ditar suas próprias políticas de transferência/sincronização, não ficando a cargo do protocolo.

Capítulo 5

Avaliação de Usabilidade

5.1 Considerações Iniciais

O estudo sobre o desenvolvimento de aplicações de captura e acesso vai além de projetar e desenvolver infraestruturas para registros e disponibilização de informações, aborda também fatores que tratam o projeto das interfaces de acesso a esses sistemas. Essas interfaces de acesso devem ser claras e organizadas, permitindo que usuários utilizem as funcionalidades dos sistemas de forma fácil, natural e eficiente.

As avaliações de usabilidade permitem identificar possíveis problemas e soluções referentes ao desenvolvimento de interfaces em sistemas computacionais. Essas avaliações utilizam técnicas que se baseiam nos processos cognitivos para verificar o quão fácil e intuitiva é a interação dos usuários com as interfaces analisadas. Diversas técnicas são propostas para auxiliar o profissional de interação na avaliação dessas interfaces. Neste trabalho são utilizadas duas técnicas de avaliação de usabilidade, uma aplicada a profissionais de interação e outra aplicada a usuários reais do sistema.

Este capítulo está estruturado da seguinte forma: na Seção 5.2 são apresentados conceitos sobre usabilidade, na qual são descritas em detalhes duas técnicas de avaliação utilizadas neste trabalho. Na Seção 5.3 são descritos os resultados obtidos a partir das avaliações com especialistas e usuários reais. Por fim, na Seção 5.4 são apresentadas as considerações finais sobre o capítulo.

5.2 Avaliação de Usabilidade

A área da computação que estuda como as pessoas interagem com os sistemas computacionais denomina-se Interação Humano-Computador (IHC), do inglês *Human-Computer Interaction* (*HCI*). Essa área de pesquisa visa estudar a forma como os usuários interagem com os sistemas computacionais, provendo técnicas para o desenvolvedor projetar interfaces que alcancem alto nível de usabilidade. O objetivo geral de estudos em IHC é alterar

a forma de abordagem dos projetos de software, onde antes os projetos eram centrados no sistema, a partir das pesquisas em IHC os projetos passam a centrar-se no usuário. A essa abordagem dá-se o nome de projeto centrado no usuário. Foram propostas técnicas em IHC para identificar quão próximas a navegação e as interfaces estão do modelo mental do usuário alvo [Motti 2009].

A usabilidade é considerada o fator que assegura que os produtos (software) sejam fáceis de usar, eficientes e agradáveis — da perspectiva do usuário. [Preece et al. 2005] definem que a usabilidade é dividida em seis metas:

- A primeira é a *eficácia*, esta é uma meta bastante geral, que determina se um sistema é bom em fazer o que se espera dele;
- A segunda meta é a *eficiência*, refere-se à maneira como o sistema auxilia os usuários na realização de suas tarefas;
- A terceira meta é a *segurança*, consiste em proteger e auxiliar o usuário na realização de tarefas evitando que situações acidentais gerem ações indesejáveis;
- A quarta é a *utilidade*, refere-se a um conjunto apropriado de funções que permite que os usuários realizem todas as tarefas da maneira que desejam;
- A quinta meta é a *capacidade de aprendizagem*, atém-se a quão fácil é aprender a usar o sistema;
- A última meta é a *capacidade de memorização*, refere-se à capacidade de lembrar como utilizar o sistema após já ter aprendido como fazê-lo.

Nos projetos centrado no usuário, o desenvolvedor deve direcionar seus esforços para o usuário, fornecendo estruturas de interfaces que sejam claras e simples de usar. Um projeto centrado no usuário deve deixar claro o que ele pode ou não fazer, como deve interagir com o sistema e qual resultado ele obterá [Hix e Hartson 1993].

De acordo com [Hix e Hartson 1993] um projeto centrado no usuário é um processo interativo que considera *design*, implementação e avaliação. O processo de avaliação de interfaces deve ser uma tarefa muito bem projetada, para mostrar as reais necessidades do sistema. É essencial que os projetistas de interfaces saibam porque é importante avaliar, o que avaliar e quando avaliar [Preece et al. 2005].

Para avaliação de interfaces são utilizadas técnicas de avaliação que se baseiam nos processos cognitivos e nos princípios de usabilidade. Os principais objetivos das técnicas de avaliação são: avaliar a acessibilidade e identificar eventuais problemas específicos com os sistemas, além de avaliar a experiência de interação dos usuários [Dix et al. 2003].

Várias técnicas foram propostas para avaliar a usabilidade em sistemas computacionais. Algumas dessas técnicas são desempenhadas por especialistas na área de projeto de interação, outras são desenvolvidas com participação do usuário final. As avaliações

realizadas pelos profissionais, geralmente, são aplicadas durante o processo de desenvolvimento das interfaces. Essas técnicas visam descobrir possíveis erros e falhas que podem ocorrer desde o desenvolvimento até a utilização pelo usuário. As principais técnicas de avaliações realizadas pelos profissionais são: percurso cognitivo e avaliação heurística. Por outro lado, existem as avaliações que exigem a participação do usuário, essas, em geral, utilizam métodos experimentais e empíricos, métodos de observação e consultas. As avaliações com participação dos usuários tendem a acontecer após o período de implementação do sistema, em que, pelo menos, protótipos já estão desenvolvidos. *Think aloud*, avaliação cooperativa e entrevistas são algumas técnicas de avaliação de usabilidade com a participação do usuário [Dix et al. 2003].

No projeto aqui reportado foram empregadas duas técnicas para avaliar a usabilidade das interfaces desenvolvidas, a primeira foi a avaliação heurística, realizada por profissionais da área de interação; a segunda técnica foi a aplicação de questionários de avaliação, que contou com a participação dos usuários.

5.2.1 Avaliação Heurística

A avaliação heurística [Nielsen 1993] é uma técnica de inspeção de usabilidade que utiliza um conjunto de princípios de usabilidade, conhecidos como heurísticas, para avaliar os elementos das interfaces de um determinado sistema. Essa técnica é aplicada por um conjunto de avaliadores que examinam as interfaces, verificando se elas estão de acordo com as heurísticas. [Nielsen 1993] definiu 10 heurísticas consideradas importantes para avaliação de interfaces. São elas:

- **Visibilidade do *status* do sistema:** O usuário deve estar ciente do estado atual do sistema; o que está acontecendo; se houve sucesso ou falha.
- **Compatibilidade do sistema com o mundo real:** A linguagem do sistema deve ser clara e simples para o usuário. Devem ser utilizadas palavras, frases e conceitos familiares aos usuários.
- **Controle e liberdade do usuário:** O sistema deve permitir que usuários saiam de lugares onde não esperariam estar. Devem ser disponibilizadas opções para o usuário retornar para onde estava antes no sistema.
- **Consistência e padrões:** Utilizar o mesmo padrão de ícones, fontes, cores e sequências de interação. As maneiras de realizar ações semelhantes devem ser consistentes.
- **Prevenção de erros:** O sistema deve evitar que um erro aconteça. É melhor prevenir do que exibir uma mensagem de erro.
- **Ajudar os usuários a reconhecer, diagnosticar e corrigir erros:** As mensagens de erro devem utilizar linguagem simples para descrever o problema e sugerir

maneiras de resolvê-lo.

- **Reconhecer em vez de relembrar:** Identificação fácil e simples dos recursos do sistema. Os objetos, as ações e opiniões devem estar sempre visíveis.
- **Flexibilidade e eficiência no uso:** Disponibilizar mecanismos para usuários mais avançados acessarem recursos mais facilmente. As opções mais acessadas devem estar em locais de fácil acesso e podem ser acessadas de diferentes formas.
- **Estética e *design* minimalista:** Não devem ser utilizadas informações desnecessárias e irrelevantes. Quanto menos palavras forem utilizadas para descrever uma determinada funcionalidade, melhor para o usuário.
- **Ajuda e documentação:** Devem ser oferecidas ajudas para auxiliar usuários na realização de suas tarefas.

Para a avaliação heurística, é recomendado que pelo menos cinco especialistas trabalhem na avaliação de um determinado sistema. Segundo [Nielsen 1993], cinco avaliadores já conseguem identificar cerca de 75% dos problemas de usabilidade.

A avaliação deve ser conduzida pelos avaliadores sobre cada interface do sistema. Somente depois de concluídas todas as avaliações é que os resultados devem ser analisados e considerados para melhorias nas interfaces do sistema. Ao final de uma avaliação heurística, o avaliador apresenta um relatório com uma lista de problemas de usabilidade encontrados na interface e uma justificativa para eles. Os problemas encontrados durante a avaliação serão classificados por níveis de severidade. Sugestões também podem ser propostas pelos avaliadores.

5.2.2 Avaliação por Questionários

Os questionários de avaliação são uma técnica bem estabelecida para a coleta de dados e opiniões de usuários [Preece et al. 2005]. Questionários podem ser elaborados para diversos domínios, inclusive para opiniões de usuários sobre usabilidades de interfaces de software. Os questionários podem ter diferentes tipos de questões. Como apresentado em [Motti 2009], os questionários possuem duas grandes vantagens: a primeira é a possibilidade de solicitar ao usuário melhorias para o sistema; a segunda vantagem é a possibilidade de comparar a avaliação de participantes diferentes, já que todos recebem o mesmo conjunto de perguntas. [Filardi e Traina 2008] acrescentam mais duas vantagens na utilização de questionários: podem ser coletados tanto dados qualitativos quanto quantitativos, e o questionário dá ao avaliador a possibilidade de adequar as questões ao seu cenário.

Ao elaborar um questionário, uma série de questões devem ser analisadas. Questionários longos podem dificultar o preenchimento pelo usuário, mas, para resolver isso,

as questões podem ser agrupadas logicamente para auxiliar o usuário no seu preenchimento. [Preece et al. 2005] sugerem algumas recomendações para o projeto de um questionário: faça perguntas claras e objetivas, ofereça várias opções de respostas, pense sobre a ordem das perguntas, evite perguntas múltiplas e complexas, ofereça instruções sobre o preenchimento do questionário. Podem ser aplicados diferentes tipos de perguntas e respostas em um questionário. Tipos diferentes de perguntas exigem tipos diferentes de respostas. Um tipo de resposta muito utilizado é a marcação de quadrinhos, pode ser estruturado desde marcação de Sim ou Não, até escalas que determinem intervalos de valores.

Segundo [Filardi e Traina 2008], existem diversos questionários disponíveis na literatura que contemplam aspectos de usabilidade de software, cada um com propósito diferente. Os principais são:

- O *QUIS*¹ (*Questionnaire for User Interaction Satisfaction*), um questionário estruturado de forma modular e organizado hierarquicamente. Foi projetado para medir a satisfação global do sistema, abordando onze fatores específicos de interface;
- O *WAMMI*² (*Website Analysis and MeasurMent Inventory*), é um questionário exclusivo para avaliação de *Websites*. Auxilia no cumprimento de metas de usabilidade por meio de análises das reações do usuário sobre a facilidade do uso do sistema.
- O *SUS*³ (*System Usability Scale*), é um questionário simples, que aborda uma visão global de estimativas subjetivas de usabilidade. Compõe-se de dez questões no formato de escala *Likert*. É um questionário que provê alto nível de subjetividade e é, frequentemente, usado para comparação de usabilidade de sistemas.
- O *SUMI*⁴ (*Software Usability Measurement Inventory*), um questionário desenvolvido pelo *Human Factors Group (HFC)* da University College. O SUMI é um questionário de avaliação utilizado para medir a satisfação do usuário durante sua interação com o sistema. Este é um questionário composto por 50 questões correspondentes a três níveis: “concordo”, “indeciso” ou “não concordo”. Os itens avaliados pelo SUMI estão classificados em cinco fatores: a *afinidade*, que analisa reações emocionais do usuário; a *eficiência*, que avalia o nível no qual o software ajuda os usuários em seu trabalho; a *eficácia*, que mede o quanto o software é de fácil compreensão e se os recursos de ajuda ajudam os usuários; *controle*, que avalia o quanto os usuários se sentem no controle do software e o *aprendizado*, que mede a velocidade e facilidade do usuário em aprender novas funções do software.

Nas próximas seções serão apresentados e discutidos os resultados das avaliações aplicadas ao *Classroom eXperience (CX)*.

¹<http://lap.umd.edu/QUIS>

²<http://www.wammi.com>

³<http://www.mindd.com>

⁴<http://sumi.ucc.ie>

5.3 Avaliação de Usabilidade do *Classroom eXperience*

Como já citado, para a análise de usabilidade dos protótipos implementados, foram empregadas duas técnicas de avaliação: a avaliação de usabilidade heurística e a avaliação de usabilidade por questionários. A avaliação heurística foi aplicada com o intuito de encontrar uma posição, do ponto de vista de profissionais da área de IHC, sobre as interfaces do *CX* antes de disponibilizá-las aos usuários finais. Os questionários foram aplicados a usuários finais após alguns dias de efetiva utilização do sistema.

5.3.1 Resultado da Avaliação Heurística

A avaliação heurística foi aplicada a cinco avaliadores que possuem conhecimentos na área de IHC. Essa quantidade de avaliadores permite encontrar 75% dos problemas de usabilidade existentes nas interfaces [Nielsen 1993]. Foi conduzido que avaliassem as interfaces com base nas dez heurísticas propostas por Nielsen, e, para cada problema encontrado, poderiam classificar o quão grave é esse problema e sugerir melhorias para essa heurística. Para cada problema, deveria ser atribuído um nível de severidade de acordo com os valores:

- 0: Eu não concordo que isto seja um problema de usabilidade;
- 1: Problema cosmético: Não precisa ser resolvido, a menos que haja tempo extra para resolvê-lo;
- 2: Problema de usabilidade baixo: Deve ser resolvido com baixa prioridade;
- 3: Problema de usabilidade grave: Importante ser resolvido, de preferência com alta prioridade;
- 4: Problema catastrófico: Deve ser resolvido antes do produto ser liberado.

Os resultados da análise da avaliação heurística apontaram algumas falhas nas interfaces que foram classificadas em seis categorias: (i) *Linguagem*; (ii) *Flexibilidade*; (iii) *Apresentação*; (iv) *Liberdade*; (v) *Prevenção de erros*; (vi) *Ajuda e documentação*. A Tabela 5.1 apresenta as principais sugestões definidas pelos avaliadores para os problemas encontrados em cada categoria.

Os resultados da avaliação heurística foram satisfatórios, diversos problemas que poderiam dificultar a interação do usuário com o sistema foram detectados. Na categoria de *linguagem*, são expostas recomendações que dizem respeito às mensagens que o sistema apresenta ao usuário. Os avaliadores apontaram que essas mensagens devem ser mais claras, utilizar uma linguagem mais detalhada e sempre manter um padrão. Na categoria de *flexibilidade*, são indicadas algumas sugestões de como deixar as interfaces mais dinâmicas e automatizadas para o usuário. Esta categoria envolve, por exemplo, a criação de

Linguagem	A linguagem deve ser mais detalhada Padronizar todas as mensagens informadas ao usuário
Flexibilidade	Retornar automaticamente após sucesso nas alterações Permitir utilização de atalhos Identificar de forma automática o tipo do usuário no <i>login</i> (professor ou aluno)
Apresentação	Alguns itens de diferentes tamanhos Utilizar cores diferentes para sucesso e falha Ordenar as informações que são apresentadas nos <i>Combo-boxes</i>
Liberdade	Botão de “voltar” em cada tela
Prevenção de erros	Limitar o tamanho dos campos
Ajuda e documentação	O sistema possui pouca ajuda Criar documentação geral do sistema

Tabela 5.1: Resultados da Avaliação Heurística

atalhos pelo teclado e retorno automático após a realização de alterações. Na categoria de *apresentação*, foram apontadas três recomendações referentes à forma como os dados são apresentados ao usuários. Essas recomendações referem-se, basicamente, a forma como as informações são ordenadas e classificadas. Na categoria de *liberdade*, foi apresentada uma sugestão, adicionar um botão “voltar” em cada tela do sistema. Essa sugestão permite dar maior controle ao usuário sobre o sistema, pois, muitas vezes, ele não consegue navegar pela trilha de navegação. Na categoria *prevenção de erros*, foi sugerido que seja limitado o tamanho dos campos para inserção de dados. Por fim, na categoria de *ajuda e documentação*, foi relatado que o sistema oferece pouca ajuda e seria interessante ter uma área específica com documentação completa do sistema.

Os problemas classificados como catastróficos e graves foram solucionados antes da disponibilização aos usuários, os demais estão sendo analisados e solucionados com menor prioridade.

Nota-se que, mesmo sendo encontrados alguns problemas de usabilidade, as interfaces do sistema atenderam positivamente a muitas heurísticas, como, por exemplo: compatibilidade do sistema com o mundo real; a consistência e padrões de fontes, cores e sequência na realização de uma tarefa; a visibilidade do *status* do sistema; o sistema ajuda o usuário a reconhecer em vez de relembrar; a estética e *design* minimalista também foram pontuados positivamente.

5.3.2 Resultado da Aplicação de Questionários

Para avaliação de usabilidade pelos usuários do sistema foi utilizada a técnica de aplicação de questionários. Foi aplicada uma versão estendida do questionário SUMI, uma versão proposta por [Motti 2009]. O questionário utilizado pode ser visto no Anexo A.

15 questionários SUMI foram respondidos e analisados. Dentre esses 15, 12 foram

respondidos por alunos que possuem conhecimento com avaliações de usabilidade e os outros 3 nunca tinham entrado em contato com avaliações de usabilidade anteriormente. Algumas conclusões mostram que:

- a maioria dos usuários respondeu que a documentação do software não é muito informativa;
- os usuários preferem utilizar as funções mais conhecidas;
- 40% dos usuários ficaram indecisos em responder se o CX ajuda a superar os problemas que tiveram durante a interação, e 20% acham que ele não ajuda;
- somente 47% dos usuários concordaram que o sistema possui uma apresentação bem atraente.

Mesmo com alguns pontos consideravelmente negativos pelos usuários, eles ainda pontuaram que:

- as instruções e mensagens ajudam na interação com o sistema;
- somente um usuário respondeu que o software parou de repente;
- 93% dos usuários acharam fácil aprender a usar o CX;
- 93% dos usuários se sentem no controle quando usam o sistema;
- Nenhum dos usuários encontrou problema de inconsistência no CX;
- 67% dos usuários concordam que as necessidades dos usuários foram levadas em consideração, ao passo que 20% ficaram indecisos.

Pôde-se perceber um impacto positivo do CX para os alunos a partir dos seguintes dados:

- todos os usuários responderam que recomendariam o CX para seus colegas;
- 80% dos usuários disseram que gostam de interagir com o CX, sendo que 20% não tinham certeza sobre isso;
- 87% dos usuários apontaram que é satisfatório trabalhar com o sistema;
- 80% dos usuários responderam que se sentem motivados a interagir com o CX;
- 93% dos usuários relataram que gostariam de usar o sistema diariamente.

Esses dados mostram que existem algumas melhoras que podem ser feitas no sistema, como uma apresentação mais atraente, melhoras nas documentações e nos procedimentos que auxiliem os usuários em caso de erros; por outro lado, o sistema se mostrou muito eficaz, pois a maioria dos usuários pontuaram positivamente sobre questões relacionadas à utilização do mesmo, por exemplo, todos os usuários recomendariam o CX para seus colegas, também, uma grande maioria disse que gostaria de utilizar o sistema diariamente.

As respostas completas aos questionários SUMI podem ser vistas a partir dos gráficos apresentados no Apêndice B.

5.4 Considerações Finais

Neste capítulo foi apresentado o estudo de usabilidade desenvolvido sobre as interfaces do CX. Este estudo contou com a aplicação de duas técnicas para verificar a clareza, organização e facilidade de uso das interfaces do sistema. Foram utilizados profissionais da área de interação e usuários reais para essas avaliações.

A primeira avaliação aplicada foi a avaliação heurística, que contou com a participação de cinco profissionais da área de IHC. Os resultados dessa avaliação apontaram algumas falhas nas interfaces, que os avaliadores classificaram em seis categorias — *linguagem, flexibilidade, apresentação, liberdade, prevenção de erros, ajuda e documentação* — descrevendo algumas sugestões em cada uma dessas. Os problemas mais graves foram solucionados antes da disponibilização do sistema para uso real. Mesmo encontrando alguns problemas de usabilidade nas interfaces, o CX está adequado com a maioria das heurísticas, mostrando ser muito claro e fácil de utilizar.

A outra avaliação foi por meio de aplicação de questionários a usuários reais. Quinze usuários responderam o questionário SUMI sobre o sistema, dessa forma, pôde-se chegar à conclusão que o sistema atende bem a questões de usabilidade, mesmo possuindo alguns detalhes que poderiam ser melhorados. Por meio dos resultados, pôde-se perceber também que os usuários gostam de interagir com o sistema e aprovaram a sua utilização.

Capítulo 6

Trabalhos Relacionados

6.1 Considerações Iniciais

Trabalhos em computação ubíqua por sua natureza são de caráter interdisciplinar, ou seja, contemplam diversas frentes de estudo para o desenvolvimento de uma determinada pesquisa. Os trabalhos relacionados apresentados neste capítulo objetivam mostrar uma visão geral das pesquisas desenvolvidas para o contexto de computação ubíqua, aplicadas a ambientes de ensino.

Os trabalhos relacionados que serão apresentados podem ser divididos em três temas: captura e acesso (C&A) de experiências, modelos de comunicação para aplicações de C&A e, por fim, personalização de conteúdo. O primeiro aborda diferentes propostas de sistemas para atividades de C&A aplicadas em diversos cenários; o segundo, diferentes abordagens para transferência e armazenamento das informações capturadas; e o terceiro, aspectos para personalização da interface de acesso aos usuários.

A organização dos tópicos tratados neste capítulo é a seguinte: na Seção 6.2, serão mostrados os trabalhos relacionados à C&A; na Seção 6.3, os trabalhos referentes a modelos de comunicação para computação ubíqua; na Seção 6.4, trabalhos com tema de pesquisa em personalização de conteúdo; finalmente, na Seção 6.5, são apresentadas as considerações finais sobre o capítulo.

6.2 Captura e Acesso de Experiências

O desenvolvimento de aplicações para captura de experiências vividas pelos usuários tem sido muito explorado como tema de pesquisa, não somente aplicadas a ambientes educacionais, mas em diversos outros cenários [Chiu et al. 2001, He et al. 2003, Kunz et al. 2010]. As aplicações desenvolvidas para cenários de ambientes educacionais ganharam, no passar dos anos, muita força, pois foi perceptível o aumento da produtividade e interatividade dentro da sala de aula [Zeng et al. 2010, AiHua 2010].

Um exemplo bem consolidado de aplicação de C&A aplicado a ambientes de educação pode ser observado no *Terakoya* [Nishiuchi et al. 2010], um sistema de ensino que auxilia alunos a estudar em qualquer local na escola (Ver Figura 6.1). Nesse sistema, professores e alunos interagem por meio de *Tablet PCs*, e todas as informações descritas pelo professor no seu dispositivo são apresentadas aos alunos em tempo real ou posteriormente, de acordo com a disponibilidade do aluno. Essa aplicação pode trabalhar de duas formas: *collaboration mode*, em que os alunos são livres para criar grupos de discussões, sendo que as informações escritas são controladas pelo professor; e *free mode*, no qual os alunos podem escrever sobre o material e enviar ao professor. Nesse caso, a informação escrita pelo aluno é vista somente pelo seu professor. Todas as informações capturadas são armazenadas em servidores que repassam o conteúdo sob demanda aos alunos. Essa implementação dá suporte a múltiplos servidores realizando a troca de informações entre si, contudo, os dispositivos dos alunos não funcionam quando estão em redes distintas dos PCs dos professores. Uma das vantagens da abordagem de comunicação proposta neste trabalho, é permitir que serviços comuniquem entre si, mesmo estando em redes diferentes.



Figura 6.1: Professor e aluno interagindo entre si através do sistema *Terakoya* [Nishiuchi et al. 2010].

MSRLCS [Zhang et al. 2008], um sistema bastante robusto que suporta captura, armazenamento, apresentação ao vivo, visualização e consultas para diversos ambientes — reuniões, palestras ou aulas. No momento da captura, todos os fluxos são armazenados em disco, possibilitando, além da apresentação ao vivo, a visualização posterior do conteúdo previamente gravado. O conteúdo produzido é acessível através de *Browsers Web*. Esse sistema utiliza um módulo de captura chamado *iCam2*, o qual grava todas as informações de áudio, vídeo e anotações sobre *slides*. São utilizados diversos microfones e câmeras, que registram todos os eventos produzidos durante esses acontecimentos.

Uma das características dessas aplicações de C&A é a utilização de diferentes dispositivos integrados ao ambiente para prover mecanismos de captura de forma transparente ao usuário. Muitas dessas aplicações utilizam lousas eletrônicas para o processo de captura

da informação ministrada. Exemplo de utilização desse recurso pode ser observado no *CollaBoard* [Kunz et al. 2010], um sistema que permite conferência entre colaboradores remotos, o qual realiza a captura de artefatos de áudio, vídeo e anotações sobre a lousa. A aplicação da lousa eletrônica utilizada no *Classroom eXperience (CX)* difere da proposta do *CollaBoard* por utilizar uma abordagem diferente para transferência das informações capturadas. No *CX* todos os traços capturados durante uma sessão são sincronizados com outras mídias, armazenados localmente e transferidos para um serviço de armazenamento *peer-to-peer* (P2P) somente após o término dessa sessão. Por outro lado, no *CollaBoard*, essas informações são transferidas periodicamente, pois é necessário prover aos usuários uma noção de tempo real das informações que estão sendo produzidas. O *CX* não suporta interação em tempo real, mas o modelo de comunicação é mais versátil que o proposto em *CollaBoard*. Outra diferença entre esses dois trabalhos é o fato do *CollaBoard* utilizar softwares de terceiros para captura do áudio e vídeo (Skype¹ e ConferenceXP²), o que não acontece no *CX*.

Semelhante ao *CollaBoard* e *MSRLCS*, a aplicação *Tele-Board* [Gumienny et al. 2011] apoia colaboração remota entre grupos de pessoas. Esse sistema é composto por um conjunto de artefatos de software que permite aos usuários fazer anotações por meio de *Tablet PCs*, *smartphones* ou diretamente na lousa (ver Figura 6.2). Todas as informações são sincronizadas automaticamente e propagadas para todos os usuários participantes da sessão. O sistema consiste de quatro software: uma aplicação Web, um cliente para a lousa, um bloco para anotações e um componente servidor (*Server*). Todos os componentes, exceto a aplicação Web, comunicam usando um protocolo XMPP (*Extensible Messaging and Presence Protocol*), um protocolo baseado em XML para tratar e rotear as mensagens. Uma das grandes vantagens proporcionadas por esse sistema é a possibilidade de reconhecimento de expressões faciais dos participantes, o que não é suportado no *CX*, em contrapartida, a infraestrutura de comunicação proposta no *CX* é mais eficiente, pois não é necessário arquiteturas centralizadas.

Sistemas de C&A para o domínio educacional nem sempre utilizam lousas eletrônicas para realizar a captura das informações, muitos utilizam câmeras de alta resolução para capturar o conteúdo em lousas tradicionais, como é o caso de *ReBoard* [Branham et al. 2010], um sistema que é capaz de capturar imagens de uma lousa e torná-las acessíveis por meio de um conjunto de ferramentas centradas no acesso do usuário. Usuários podem acessar o conteúdo capturado por meio da Web ou de um dispositivo chamado *Chumby*³. Com a *ReBoard*, os usuários podem imprimir, enviar *email*, realizar *downloads* e compartilhar as imagens produzidas durante uma sessão de captura. Diferente do sistema proposto neste trabalho (*CX*), que utiliza a própria lousa para capturar as imagens, a

¹<http://www.skype.com>

²<http://research.microsoft.com/en-us/projects/conferencexp/>

³Chumby. www.chumby.org

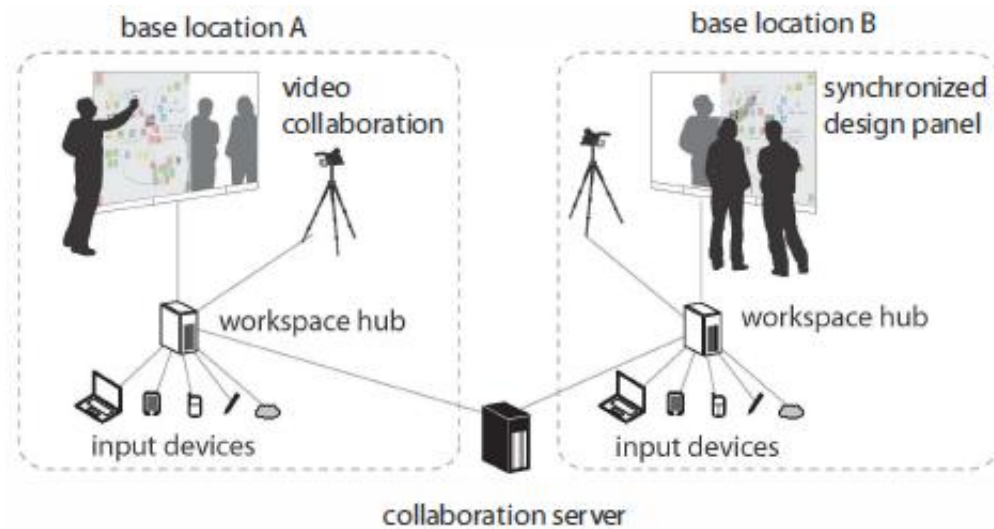


Figura 6.2: Protótipo do sistema *Tele-Board* [Gumienny et al. 2011].

ReBoard realiza essa captura mediante câmeras de alta resolução instaladas no ambiente. Outra diferença significativa entre os dois trabalhos pode ser observada nos conteúdos resultantes de uma sessão de captura; enquanto o *CX* tem a capacidade de gerar o conteúdo dinamicamente, de acordo com o contexto de acesso de cada usuário, a *ReBoard* se limita a gerar somente conteúdo de forma estática, sem se adaptar ao ambiente de acesso.

[Dickson et al. 2006, Dickson et al. 2008] propõem abordagens para que a captura ocorra de forma muito transparente, evitando que o professor altere seu estilo de comportamento dentro de sala da aula. Outra característica deste sistema é a capacidade de capturar informações em lousas com altas dimensões, não se limitando a pequenos *displays*. Diferente da abordagem explorada no *CX*, que captura fluxos contínuos das informações, em [Dickson et al. 2008] é realizada a captura somente de eventos considerados significativos, não sendo necessário realizar a captura completa da sessão. O sistema captura eventos de áudio, vídeo e informações escritas sobre a lousa, gerando, como apresentação ao usuário, um vídeo em *Flash*⁴ com as anotações que foram realizadas durante a aula. Para classificação dos eventos significativos, os autores empregam um algoritmo que compara a imagem atual com imagens anteriores para identificar se houve mudança no conteúdo exibido. Segundo os autores, esse processo reduz, significativamente, a quantidade de imagens que são armazenadas, não prejudicando o processo de aprendizagem devido as imagens descartadas. O *CX* utiliza uma abordagem de captura contínua, pois acredita-se que é mais eficiente e proveitoso registrar todos os instantes de tempo de uma sessão que somente alguns eventos isolados.

Outro sistema que faz o uso de câmeras de alta resolução para a captura das informações na lousa é proposto por [He e Zhang 2007]. Nesse sistema são utilizados recursos de extração de imagens para extrair somente as informações escritas na lousa, eliminando

⁴<http://get.adobe.com/br/flashplayer/>

todas as imagens não necessárias — imagem do professor, imagem da sala, entre outras. Essa técnica para captura é interessante por não precisar de equipamentos extras para captura de informação, tornando assim a utilização do sistema completamente transparente ao usuário. A desvantagem dessa proposta é a necessidade de alto poder de processamento para realizar todas as segmentações e extrações de características das imagens capturadas.

A Tabela 6.1 apresenta comparações entre os sistemas apresentados acima e o *CX*.

	Modo de trabalho	Forma de comunicação	Forma de apresentação	Disp. para anotações	Possui avaliação
<i>Terakoya</i>	Colaborativo	Sint. de transferência	Tablet	Tablet	Sim
<i>MSRLCS</i>	Colaborativo	Protocolo	Web	Lousa tradicional	Não
<i>CollaBoard</i>	Colaborativo	Sint. de transferência	Diretamente na lousa	Lousa eletrônica	Sim
<i>Tele-Board</i>	Colaborativo	Protocolo	Web ou na lousa	Lousa eletrônica	Não
<i>ReBoard</i>	Individual	Sint. de transferência	Web ou Chumby	Tablet, smartphone ou Lousa elet.	Não
[Dickson et al. 2008]	Individual	Sint. de transferência	Video em Flash	Lousa tradicional	Não
[He e Zhang 2007]	Individual	Sint. de transferência	Web	Lousa tradicional	Não
<i>Classroom eXperience</i>	Individual	Protocolo	Web e TV	Lousa eletrônica	Sim

Tabela 6.1: Comparação entre sistemas de captura e acesso

6.3 Modelos de Comunicação para Aplicações de Captura e Acesso

A forma como os dados gerados a partir de uma sessão de captura trafegam pela rede e são armazenados nos serviços de armazenamento é um dos grandes desafios da computação ubíqua [Truong e Hayes 2009]. Existem diversas abordagens de pesquisa que propõem modelos de comunicação para esse fim. Algumas estruturam a rede em arquitetura cliente/servidor (C/S) [Nishiuchi et al. 2010, Kunz et al. 2010, Giliberti et al. 2011]. Outras, por sua vez, preferem uma comunicação mais descentralizada, em que cada nó na rede pode atuar como cliente e servidor dinamicamente; esses utilizam uma arquitetura P2P [Kameas et al. 2002, Barolli e Xhafa 2011, Xie et al. 2010]. Outros ainda,

preferem abordagens *publish/subscriber* [Pimentel et al. 2007]. Independente da forma como a comunicação é estruturada, o principal objetivo dessas pesquisas é prover mecanismos eficientes para armazenar os dados de forma a proporcionar alta disponibilidade das informações produzidas.

Certas aplicações de C&A têm seu modelo de comunicação estruturado para trabalhar em modo síncrono, ou seja, ao passo que as informações vão sendo capturadas, elas serão distribuídas para os usuários. Exemplos de modelos de comunicação que utilizam uma estrutura de comunicação síncrona podem ser vistos em [Nishiuchi et al. 2010, Kunz et al. 2010]. [Nishiuchi et al. 2010] propõem uma arquitetura C/S, em que um servidor é responsável pela distribuição do conteúdo pela rede, enquanto clientes são responsáveis por consultar as notas de aula e armazenar o material para o aluno. [Kunz et al. 2010] também propõem um mecanismo de comunicação para aplicação de C&A de forma C/S. Semelhante ao [Nishiuchi et al. 2010], nesse modelo de comunicação todos os artefatos capturados são enviados para um servidor responsável por retransmitir essas informações para os demais clientes. Vale ressaltar que nesses dois trabalhos a comunicação foi estruturada dessa forma para prover noção de tempo real entre as aplicações, o que diferencia da abordagem proposta pelo CX.

Ainda no contexto de aplicações que utilizam estruturas C/S, pode-se citar o trabalho proposto em [Giliberti et al. 2011] no qual todas as informações capturadas são enviadas periodicamente a um servidor na rede, que fica responsável por armazenar esse conteúdo. Esse é um sistema baseado em lousa eletrônica e *Tablet PCs* para o registro de informações educacionais para estudantes com necessidades especiais. Existem dois modos em que o sistema pode trabalhar: no primeiro modo, as informações são acessadas de forma síncrona, todos os alunos se conectam a uma VNC (*Virtual Network Computing*) em que fazem autenticação no sistema para ter acesso às informações passadas pelo professor; no segundo modo de funcionamento, o acesso às informações acontece posteriormente a sua captura, onde as aulas capturadas ficam armazenadas em um único repositório na rede. Esse segundo modo de funcionamento assemelha-se com a proposta aqui apresentada, pois, em ambos, os dados são capturados e disponibilizados posteriormente para os usuários; a diferença é que no CX é utilizada uma abordagem P2P, sendo possível existir diversos serviços de armazenamento, em que estes realizam sincronização de conteúdo entre si. Já na abordagem proposta por [Giliberti et al. 2011] existe um único serviço de armazenamento disponível na rede (Ver Figura 6.3), caso esse serviço pare de funcionar, toda a infraestrutura estará comprometida.

Algumas abordagens para comunicação de dados no domínio de educação procuram utilizar serviços de *broadcast* para disponibilizar o conteúdo pela rede. Esse é o caso de *PPClass* [Xie et al. 2010], um sistema para ensino que permite ao público acessar aulas em alta qualidade via internet. A arquitetura proposta foi organizada utilizando uma abordagem P2P *Streaming*, em que cada *peer* pode servir outros *peers* com o conteúdo

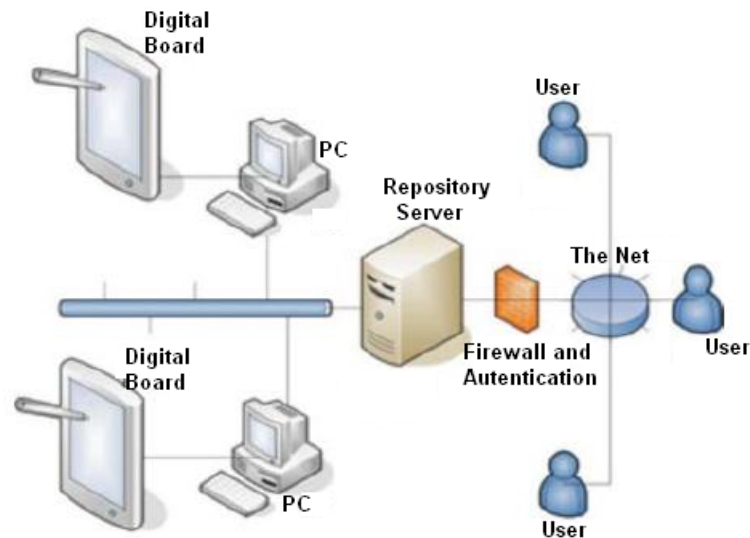


Figura 6.3: Disposição da infraestrutura de rede do sistema proposto por [Giliberti et al. 2011].

já recebido. Segundo os autores, a grande vantagem dessa abordagem está no custo, pois é possível suportar um grande número de usuários com uma infraestrutura não tão sofisticada.

Muitas pesquisas focam em propostas para conectar objetos do cotidiano das pessoas, como: mesa de estudos, lâmpadas de mesa, coleção de livros, entre outros. Em *eComP* [Kameas et al. 2002] é proposto um módulo que organize a rede de forma descentralizada, baseado em mensagens XML para realizar a comunicação entre os *peers* da rede. Como em *eComP*, existem outras pesquisas que exploram o mecanismo de comunicação na rede através de documentos XML, como é o caso do *JXTA-Overlay* [Barolli e Xhafa 2011], um *middleware* P2P baseado na plataforma JXTA para sistemas distribuídos e colaborativos. Esta proposta permite a integração de dispositivos finais, como sensores e computadores pessoais/móveis, provendo transparência e segurança para compartilhar, contribuir e controlar recursos disponíveis. *JXTA-Overlay* abrange um conjunto de operações primitivas: descoberta de *peers*, alocação de recursos, compartilhamento de dados/arquivos, descoberta e transmissão, comunicação instantânea, entre outros serviços. Mesmo apresentando uma estrutura semelhante ao CX, no *JXTA-Overlay*, o aplicativo deve conhecer o identificador do *peer* que ele deseja se conectar, por outro lado, no mecanismo de comunicação ora proposto não requer o conhecimento, a priori, do identificador do *peer*.

6.4 Personalização e Recomendação de Conteúdo

Ambientes de C&A produzem grandes volumes de informações devido a sua característica de capturar atividades cotidianas das pessoas. Esses conteúdos devem ser apresen-

tados aos clientes de forma clara e organizada, permitindo que eles possam navegar por essas informações sem grandes esforços.

Um dos grandes desafios da área de C&A está na capacidade de prover mecanismos eficientes para selecionar o melhor conteúdo a ser apresentado ao usuário [Truong e Hayes 2009, Kientz 2012]. Diversos projetos aplicados em ambientes educacionais tratam especificamente da fase de acesso da informação, em que são desenvolvidos mecanismos para classificar e organizar todo o conteúdo a ser apresentado [Zhuhadar et al. 2009, Wang e Wu 2011, Karger et al. 2008].

Em [Zhuhadar et al. 2009] é exposto um trabalho que busca personalizar a entrega de recursos educacionais de forma eficiente aos alunos. Os autores utilizam recursos providos pela Web 2.0 para personalizar o conteúdo que melhor se adapte aos usuários. O sistema combina funcionalidades de personalização de conteúdo com extração de dados de alunos para classificá-los em grupos. Uma vantagem significativa do CX sobre a proposta de [Zhuhadar et al. 2009] é a capacidade de tratamento das informações a partir do contexto de acesso do usuário. O CX tem a possibilidade de apoiar no contexto de acesso do usuário, para, a partir daí, recomendar-lhe um determinado conteúdo baseado nos seus interesses.

Em [Wang e Wu 2011] é mostrado um robusto ambiente de ensino que emprega tecnologias de *context-aware* e algoritmos de recomendação de conteúdo para auxiliar no aprendizado dos alunos. Entre as diversas características do sistema, podem-se citar o uso de tecnologia RFID (*Radio Frequency IDentification*) para prover funcionalidades de *context-aware*, e o uso de padrões de especificação SCORM (*Sharable Content Object Reference Model*) para permitir reusabilidade e suporte a multiplataforma. O sistema de recomendação analisa as preferências de cada aluno no seu perfil de aprendizado pessoal e recomenda os recursos de aprendizado que melhor se adaptem às suas necessidades. A grande vantagem do CX sobre essa proposta é a possibilidade da recuperação dos conteúdos poder acontecer diretamente no banco de dados, não sendo necessário que a aplicação fique responsável por essa tarefa.

Dentre os sistemas que utilizam contexto para recomendação do conteúdo, pode-se citar também a proposta de [Wu et al. 2008], um sistema de recomendação de objetos de aprendizado, que apresenta o material que mais corresponda aos interesses do aluno, mediante seu contexto. Uma das características desse sistema está no fato de a recomendação acontecer a estudantes individuais, e não a grupos.

Semelhante à abordagem do CX, em [Karger et al. 2008] é apresentada uma infraestrutura para selecionar recursos de aprendizado baseados em consultas com suporte a preferências. Esse é um sistema de gestão de conteúdos educacionais, em que o aluno informa as suas preferências — cursos a distância ou presencial, quais são os dias de aula, se contém ou não avaliação, entre outras — e o sistema informa o curso que seja mais adequado às preferências de cada usuário. Como em [Zhuhadar et al. 2009], uma das

limitações deste trabalho está no fato de o sistema não tratar do contexto do usuário, ou seja, a forma como o aluno está fazendo o acesso ao sistema não interfere no resultado da consulta. No trabalho aqui apresentado, *CX*, caso o usuário não informe nenhuma preferência, o sistema seleciona uma previamente informada baseado em seu contexto de acesso.

Uma abordagem para a recomendação de conteúdo muito usada é a utilização de tecnologias de Web semântica. Em ELENA [Dolog et al. 2004], é apresentado um sistema de personalização de conteúdo educacional baseado em tecnologias de Web semântica e desenvolvido sobre diversos *Web-services*. As tecnologias de Web semântica foram desenvolvidas para representar conhecimento que integra repositórios e serviços distribuídos sem a necessidade de controle centralizado. Outra abordagem encontrada na literatura para personalização de conteúdo é a consulta personalizada, em [Spyratos e Christophides 2006] é apresentado um *framework* para personalização de consultas em uma biblioteca digital de acordo com as preferências de cada usuário. As principais funções desta abordagem estão na restrição da quantidade de tuplas retornadas para o usuário e a ordem que essas tuplas serão apresentadas. Todo o processo de personalização das consultas ocorre de forma *online*. Uma consulta personalizada Q é definida como um conjunto de três informações, uma consulta habitual (Ex. selecionar todos os documentos de Java e *QuickSort*), um inteiro k para representar a quantidade máxima de tuplas a serem retornadas (Ex. 5) e por fim, as preferências do usuário (Ex. prefiro os documentos sobre *QuickSort* a documentos sobre Java). A partir disso, o sistema retornará ao usuário as tuplas que melhor corresponderem ao conteúdo desejado.

6.5 Considerações Finais

Neste capítulo foram descritos diversos trabalhos relacionados às principais áreas abordadas neste projeto. Alguns trabalhos têm relação direta com a proposta apresentada; outros, por sua vez, não possuem uma arquitetura análoga, mas as suas características são bem semelhantes.

Inicialmente foram apresentados os trabalhos relacionados à área de C&A, em que foi possível perceber tamanha diversidade em propostas de arquitetura nesses ambientes. Na segunda seção foram apresentados os trabalhos que focam no processo de armazenamento e distribuição das informações capturadas, também foram vistas diversas abordagens para estruturação dos componentes na rede. Por fim, foram mostrados os trabalhos relacionados à fase de acesso personalizado a informações, em que foi perceptível a necessidade dessa etapa para o processo de C&A.

Capítulo 7

Conclusão

7.1 Visão Geral, Resultados e Contribuições

O trabalho ora reportado apresentou o *Classroom eXperience (CX)*, um sistema de captura e acesso (C&A) voltado para ambientes educacionais instrumentados, que provê recursos de autoria multimídia, personalização de conteúdo e mecanismos de transferência e sincronização das informações capturadas. O mesmo foi implantado em uma sala de aula instrumentada na Faculdade de Computação da Universidade Federal de Uberlândia - FACOM/UFU, onde já está em pleno uso. Com esse sistema, professores têm a possibilidade de ministrar aulas de forma mais interativa, ao passo que alunos podem concentrar mais sua atenção no conteúdo que está sendo apresentado, pois sabem que todas as informações estão sendo capturadas e serão disponibilizadas posteriormente. Isso possibilita também aos alunos, um grande volume de informações para estudo, que podem, sempre que estiverem com dúvida, rever o conteúdo apresentado em uma determinada aula.

Foram apresentados conceitos arquiteturais do CX, incluindo sua infraestrutura, com detalhamento do comportamento dos componentes e como eles interagem entre si durante todas as etapas dos processos de C&A. A proposta segue uma abordagem bem consolidada, que já foi validada por meio de muitos trabalhos desenvolvidos na área [Winer e Cooperstock 2002, Pimentel et al. 2007, Martins et al. 2011]. A infraestrutura foi implementada de forma estrutural e modular, em que componentes podem ser substituídos de acordo com a necessidade do projeto.

O CX implementa mecanismos para tratamento contextual do acesso às informações capturadas. As informações são apresentadas ao usuário de acordo com o seu contexto de acesso. Foram utilizados dois módulos para tratar o conteúdo a ser recomendado: um para a personalização “do que” é entregue ao usuário; e outro que personaliza a forma “como” esse conteúdo é apresentado.

Também foi formalizado e implementado um mecanismo para transferência e sincronização de conteúdo proveniente de aplicações de computação ubíqua. Esse mecanismo foi

implementado como um protocolo P2P que permite a transferência de conteúdo de forma transparente à aplicação. O protocolo em questão possibilita que serviços de armazenamento repliquem e sincronizem conteúdos provenientes de atividades de captura, e pode ser empregado em outros cenários de distribuição de conteúdo.

O sistema teve sua usabilidade avaliada por meio de avaliação heurística e aplicação de questionários, cujos resultados não detectaram problemas sérios na interação usuário-sistema. A avaliação heurística mostrou que mesmo apesar de haver algumas considerações que deveriam ser tomadas antes da disponibilização para os usuários, as interfaces estão, em geral, de acordo com as heurísticas de Nielsen. As interpretações dos questionários mostraram, entre outras questões, que os usuários gostaram de interagir e utilizariam o sistema na prática.

7.2 Limitações

Existem algumas limitações quanto à utilização da infraestrutura proposta em ambientes reais, mas vale ressaltar que tais limitações são relacionadas a aspectos de implementação, o que não invalida a proposta do sistema:

- A coleta de informação de contexto é atualmente realizada de modo manual, sendo necessário que o usuário informe explicitamente as três dimensões de contexto necessárias à recuperação de informação com suporte a preferências;
- Também relacionada ao suporte a preferências, outra limitação diz respeito à forma como as mesmas são cadastradas pelo usuário. A formalização de uma preferência pode ser considerada uma tarefa difícil para usuários que não tenham conhecimento da linguagem CPref-SQL;
- As interfaces do *CX* foram desenvolvidas para acesso por meio de equipamentos como *laptops* e *desktops*, que possuem telas grandes. Quando o acesso é realizado em dispositivos com telas pequenas, como por exemplos *smartphones*, as informações não ficam completamente organizadas podendo ficar até mesmo ilegíveis;
- A proposta atual não contempla como será feita a distribuição do conteúdo NCL produzido pelo sistema para *set-top boxes* Ginga. Hoje, a disponibilização desse conteúdo acontece por meio do *download* de um arquivo compactado contendo o documento NCL e as mídias capturadas, com sua reprodução ficando a cargo do usuário com a utilização de um emulador ou de sua transferência para um *set-top box*.

7.3 Trabalhos Futuros

O CX implanta uma nova frente de pesquisa na Universidade Federal de Uberlândia, permitindo que novos trabalhos sejam desenvolvidos no âmbito de captura, autoria e apresentação de conteúdo multimídia:

- Refinamento da infraestrutura de personalização de conteúdo com o aperfeiçoamento da interface de cadastro e coleta de preferências, suportando eventualmente que os mesmos ocorram de maneira automática por meio de sensoramento e de algoritmos de inferência;
- Avaliação mais detalhada do mecanismo de transferência de dados no protocolo proposto, uma vez que não foram executados testes para medir seu desempenho na rede de comunicação nem houve comparação com outros protocolos semelhantes — atividades essas que fugiam do escopo originalmente planejado para o trabalho;
- Mineração de dados sobre a informação capturada durante as aulas. Com o sistema extensivamente em uso, poderia-se, por exemplo, processar e cruzar as informações capturadas para criação de ligações hipermídia entre conteúdos relacionados;
- Personalização das interfaces de acesso para dispositivos móveis e com tela de tamanho reduzido — embora não tenha sido implementada, tal mudança é facilmente acomodada no modelo de personalização de conteúdo ora proposto, bastando no caso a definição de novas restrições de apresentação referentes a dispositivos com essas limitações;
- Integração da infraestrutura proposta com ambientes virtuais de aprendizado (AVA), como o Moodle¹, por exemplo. Nesse caso, as etapas de Pré-produção e Acesso aconteceriam por meio do AVA e as etapas de Gravação ao vivo e Pós-produção, através do CX;
- Adaptação do CX para atuar em ambientes de reuniões, o qual poderiam ser desenvolvidos mecanismos para colaboração remota entre participantes. Também poderiam ser desenvolvidas ferramentas próprias para determinados tipos de reunião, por exemplo, ferramentas para reuniões de Análise e Projeto de Software;
- Desenvolvimento e implantação de uma rede social sobre a atual interface para acesso do sistema, em que o aluno poderia estender o conteúdo capturado. Isso permitiria que os alunos enriquecessem o conteúdo originalmente apresentado pelo professor por meio de funcionalidades tais como: anotações multimodais, ranqueamento e recomendação de conteúdo, entre outras possibilidades;
- Participação de equipes multidisciplinares na análise do uso do sistema com relação a princípios psico-pedagógicos e cognitivos.

¹<http://moodle.com/>

Pode-se observar que muitos dos trabalhos futuros supracitados vão de encontro a sanar as limitações descritas na seção anterior. Estima-se que seus resultados permitam uma avaliação tanto qualitativa e quantitativa do uso do sistema, podendo, futuramente, gerar dados e informações que contribuam para o processo de aprendizado apoiado por computador.

Referências Bibliográficas

- [Abowd 1999] Abowd, G. D. (1999). Classroom 2000: an experiment with the instrumentation of a living educational environment. *IBM Syst. J.*, 38(4):508–530.
- [Abowd et al. 1996] Abowd, G. D., Atkeson, C. G., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., e Tani, M. (1996). Teaching and learning as multimedia authoring: the classroom 2000 project. In *Proceedings of the 4th ACM International Conference on Multimedia*, MULTIMEDIA '96, pp. 187–198, Boston, Massachusetts, United States, New York, NY, USA. ACM.
- [Abowd e Mynatt 2000] Abowd, G. D. e Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Trans. Computer-Human Interaction*, 7(1):29–58.
- [AiHua 2010] AiHua, Z. (2010). Study of ubiquitous learning environment based on Ubiquitous computing. In *3rd IEEE International Conference on Ubi-media Computing (U-Media)*, pp. 136–138.
- [Araújo et al. 2012] Araújo, R. D., Ferreira, H. N. M., Rosa, P. F., e Cattelan, R. G. (2012). A Redundancy Information Protocol for P2P Networks in Ubiquitous Computing Environments: Design and Implementation. In *Proceedings of 11th International Conference on Networks*, ICN '12, pp. 215–220, Saint Gilles, Reunion, Saint Gilles, Reunion. IARIA.
- [Barolli e Xhafa 2011] Barolli, L. e Xhafa, F. (2011). JXTA-Overlay: A P2P Platform for Distributed, Collaborative, and Ubiquitous Computing. *IEEE Trans. on Industrial Electronics*, 58(6):2163–2172.
- [Berque et al. 1999] Berque, D., Hutcheson, A., Johnson, D., Jovanovic, L., Moore, K., Singer, C., e Slattery, K. (1999). Using a variation of the WYSIWIS shared drawing surface paradigm to support electronic classrooms. In *Proceedings of the 8th International Conference on Human Computer Interaction*, HCI '99.
- [Black et al. 1998] Black, M. J., Berard, F., Jepson, A., Newman, W., Saund, E., Socher, G., e Taylor, M. J. (1998). The Digital Office: Overview.
- [Branham et al. 2010] Branham, S., Golovchinsky, G., Carter, S., e Biehl, J. T. (2010). Let's go from the whiteboard: supporting transitions in work through whiteboard capture and reuse. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, CHI '10, pp. 75–84, Atlanta, Georgia, USA, New York, NY, USA. ACM.

- [Bush e Wang 1945] Bush, V. e Wang, J. (1945). As we may think. *Atlantic Monthly*, 176:101–108.
- [Cattelan 2009] Cattelan, R. G. (2009). *Captura e acesso na produção, distribuição, apresentação e extensão de conteúdo multimídia*. PhD thesis, São Carlos, SP, BR. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-31082009-161619/pt-br.php>. Acesso em: 01/03/2011.
- [Chiu et al. 2001] Chiu, P., Boreczky, J., Girgensohn, A., e Kimber, D. (2001). LiteMinutes: an Internet-based system for multimedia meeting minutes. In *Proceedings of the 10th International Conference on World Wide Web, WWW '01*, pp. 140–149, Hong Kong, Hong Kong, New York, NY, USA. ACM.
- [Dey 2001] Dey, A. K. (2001). Understanding and Using Context. *Personal Ubiquitous Comput.*, 5(1):4–7.
- [Dey et al. 1999] Dey, A. K., Salber, D., Abowd, G. D., e Futakawa, M. (1999). The Conference Assistant: Combining Context-Awareness with Wearable Computing. In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers, ISWC '99*, pp. 21–28, Washington, DC, USA. IEEE Computer Society.
- [Dickson et al. 2006] Dickson, P., Adrion, W. R., e Hanson, A. (2006). Automatic Capture of Significant Points in a Computer Based Presentation. In *Proceedings of the 8th IEEE International Symposium on Multimedia, ISM '06*, pp. 921–926, Washington, DC, USA. IEEE Computer Society.
- [Dickson et al. 2008] Dickson, P. E., Adrion, W. R., e Hanson, A. R. (2008). Automatic creation of indexed presentations from classroom lectures. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '08*, pp. 12–16, Madrid, Spain, New York, NY, USA. ACM.
- [Dix et al. 2003] Dix, A., Finlay, J. E., Abowd, G. D., e Beale, R. (2003). *Human-Computer Interaction (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [Dolog et al. 2004] Dolog, P., Henze, N., Nejdl, W., e Sintek, M. (2004). Personalization in distributed e-learning environments. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, WWW Alt. '04*, pp. 170–179, New York, NY, USA, New York, NY, USA. ACM.
- [Filardi e Traina 2008] Filardi, A. L. e Traina, A. J. M. (2008). Montando questionários para medir a satisfação do usuário: avaliação de interface de um sistema que utiliza técnicas de recuperação de imagens por conteúdo. In *Proceedings of the 8th Brazilian Symposium on Human Factors in Computing Systems, IHC '08*, pp. 176–185, Porto Alegre, RS, Brasil, Porto Alegre, RS, Brasil. Sociedade Brasileira de Computação.
- [Fleck et al. 2002] Fleck, M., Frid, M., Kindberg, T., O'Brien-Strain, E., Rajani, R., e Spasojevic, M. (2002). Rememberer: A Tool for Capturing Museum Visits. In *Proceedings of the 4th International Conference on Ubiquitous Computing, UbiComp '02*, pp. 48–55, Goteborg, Sweden, London, UK, UK. Springer-Verlag.

- [Friedland et al. 2004] Friedland, G., Knipping, L., Rojas, R., e Tapia, E. (2004). Teaching with an intelligent electronic chalkboard. In *Proceedings of the 2004 ACM SIGMM Workshop on Effective Telepresence*, ETP '04, pp. 16–23, New York, NY, USA, New York, NY, USA. ACM.
- [Giliberti et al. 2011] Giliberti, E., Corradini, M., Guaraldi, G., Bertarelli, F., e Genovese, E. (2011). Alternative learning technologies for students with special educational needs. In *Proceedings of the 2011 International Symposium on IT in Medicine and Education, ITME '2011*, volume 1, pp. 633–637.
- [Greenberg et al. 1999] Greenberg, S., Boyle, M., e Laberge, J. (1999). PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. Technical report, Personal Technologies.
- [Gumienny et al. 2011] Gumienny, R., Gericke, L., Quasthoff, M., Willems, C., e Meinel, C. (2011). Tele-Board: Enabling efficient collaboration in digital design spaces. In *15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 47–54.
- [He et al. 2003] He, L. W., Liu, Z., e Zhang, Z. (2003). Why take notes? Use the whiteboard capture system. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 5, pp. V–776–9.
- [He e Zhang 2007] He, L. W. e Zhang, Z. (2007). Real-Time Whiteboard Capture and Processing Using a Video Camera for Remote Collaboration. *IEEE Transactions on Multimedia*, 9(1):198–206.
- [Hess e Campbell 2003] Hess, C. K. e Campbell, R. H. (2003). A Context-Aware Data Management System for Ubiquitous Computing Applications. In *Proceedings of the 23rd International Conference on Distributed Computing Systems, ICDCS '03*, pp. 294–301, Washington, DC, USA. IEEE Computer Society.
- [Hix e Hartson 1993] Hix, D. e Hartson, H. R. (1993). *Developing user interfaces: ensuring usability through product & process*. John Wiley & Sons, Inc., New York, NY, USA.
- [Hodges et al. 2006] Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., e Wood, K. (2006). SenseCam: A retrospective memory aid. In *Proceedings of the 8th International Conference on Ubicomp*, pp. 177–193.
- [Holzmann 1991] Holzmann, G. J. (1991). *Design and validation of computer protocols*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.
- [Kameas et al. 2002] Kameas, A., Mavrommati, I., Ringas, D., e Wason, P. (2002). eComP: an architecture that supports P2P networking among ubiquitous computing devices. In *Proceedings of the 2nd International Conference on Peer-to-Peer Computing (P2P 2002)*, pp. 57–64.
- [Karger et al. 2008] Karger, P., Olmedilla, D., Abel, F., Herder, E., e Siberski, W. (2008). What Do You Prefer? Using Preferences to Enhance Learning Technology. *IEEE Transactions on Learning Technologies*, 1(1):20–33.

- [Kientz 2012] Kientz, J. A. (2012). Embedded capture and access: encouraging recording and reviewing of data in the caregiving domain. *Personal Ubiquitous Comput.*, 16(2):209–221.
- [Kunz et al. 2010] Kunz, A., Nescher, T., e Ku andchler, M. (2010). CollaBoard: A Novel Interactive Electronic Whiteboard for Remote Collaboration with People on Content. In *International Conference on Cyberworlds (CW)*, pp. 430–437.
- [Lamming e Flynn 1994] Lamming, M. e Flynn, M. (1994). “Forget-me-not” Intimate Computing in Support of Human Memory. In *Proceedings of Friend 21: International Symposium on Next Generation Human Interface*, pp. 125–128.
- [LTSC IEEE 2002] LTSC IEEE (2002). Draft standard for learning technology - Learning Object Metadata. Technical report, IEEE Standards Department, New York. Disponível em: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf. Acesso em: 17/04/2011.
- [Mamykina et al. 2004] Mamykina, L., Goose, S., Hedqvist, D., e Beard, D. V. (2004). CareView: analyzing nursing narratives for temporal trends. In *Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pp. 1147–1150, Vienna, Austria, New York, NY, USA. ACM.
- [Martins et al. 2011] Martins, D. S., Vega-Oliveros, D. A., e da Graça Campos Pimentel, M. (2011). Automatic authoring of interactive multimedia documents via media-oriented operators. *SIGAPP Appl. Comput. Rev.*, 11(4):26–37.
- [Modahl et al. 2004] Modahl, M., Agarwalla, B., Abowd, G., Ramachandran, U., e Saponas, T. S. (2004). Toward a standard ubiquitous computing framework. In *Proceedings of the 2nd Workshop on Middleware for Pervasive and ad-hoc Computing*, MPAC '04, pp. 135–139, Toronto, Ontario, Canada, New York, NY, USA. ACM.
- [Morse e Dey 2000] Morse, D. R. e Dey, A. K. (2000). The What, Who, Where, When, Why and How of Context-Awareness? *CHI 2000*, 1(1):1–2.
- [Motti 2009] Motti, V. G. (2009). Design Centrado no Usuário de um Ambiente de Reunião Instrumentado. Mestrado em ciência da computação e matemática computacional, Universidade de São Paulo, São Carlos. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-26052009-143029/pt-br.php>. Acesso em: 01/11/2011.
- [Mukhopadhyay e Smith 1999] Mukhopadhyay, S. e Smith, B. (1999). Passive capture and structuring of lectures. In *Proceedings of the 70th ACM International Conference on Multimedia (Part 1)*, MULTIMEDIA '99, pp. 477–487, Orlando, Florida, United States, New York, NY, USA. ACM.
- [Nielsen 1993] Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [Nishiuchi et al. 2010] Nishiuchi, Y., Matsuuchi, N., Yamaguchi, T., Shiba, H., Fujiwara, K., Mendori, T., e Shimamura, K. (2010). Enhanced TERA KOYA learning system providing multi-point remote interactive lessons. In *8th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT)*, pp. 1–4.

- [Pedersen et al. 1993] Pedersen, E. R., McCall, K., Moran, T. P., e Halasz, F. G. (1993). Tivoli: an electronic whiteboard for informal workgroup meetings. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pp. 391–398, Amsterdam, The Netherlands, New York, NY, USA. ACM.
- [Pereira e de Amo 2010] Pereira, F. S. F. e de Amo, S. (2010). Evaluation of Conditional Preference Queries. *JIDM*, 1(3):503–518.
- [Pimentel et al. 2007] Pimentel, M. d. G. C., Baldochi Jr., L. A., e Cattelan, R. G. (2007). Prototyping Applications to Document Human Experiences. *IEEE Pervasive Computing*, 6:93–100.
- [Plaisant et al. 1996] Plaisant, C., Milash, B., Rose, A., Widoff, S., e Shneiderman, B. (1996). LifeLines: visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground*, CHI '96, pp. 221–227, Vancouver, British Columbia, Canada, New York, NY, USA. ACM.
- [Preece et al. 2005] Preece, J., Rogers, Y., e Sharp, H. (2005). *Design de Interação: Além da interação homem-computador*. Bookman, Porto Alegre, RS, Brasil.
- [Richter et al. 1998] Richter, H. A., Schuchhard, P., e Abowd, G. D. (1998). Automated Capture and Retrieval of Architectural Rationale. Technical report, Gvu Center, Georgia Institute of Technology. Disponível em: <http://smartech.gatech.edu/handle/1853/3485>. Acesso em 17/03/2012.
- [Silva et al. 2004] Silva, H. V. O., Rodrigues, R. F., Soares, L. F. G., e Muchaluat Saade, D. C. (2004). NCL 2.0: integrating new concepts to XML modular languages. In *Proceedings of the 2004 ACM Symposium on Document Engineering*, DocEng '04, pp. 188–197, Milwaukee, Wisconsin, USA, New York, NY, USA. ACM.
- [Spyratos e Christophides 2006] Spyrtos, N. e Christophides, V. (2006). Querying with Preferences in a Digital Library. In Jantke, K., Lunzer, A., Spyrtos, N., e Tanaka, Y. (editores), *Federation Over the Web*, volume 3847 de *Lecture Notes in Computer Science*, pp. 130–142. Springer Berlin / Heidelberg.
- [Streitz et al. 1994] Streitz, N. A., Geissler, J., Haake, J. M., e Hol, J. (1994). DOLPHIN: integrated meeting support across local and remote desktop environments and LiveBoards. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pp. 345–358, Chapel Hill, North Carolina, United States, New York, NY, USA. ACM.
- [Sun Microsystems, Inc. 2007] Sun Microsystems, Inc. (2007). Project JXTA v2.5: Java Programmer's Guide. Disponível em: <http://java.net/projects/jxta>. Acesso em: 20/07/2011.
- [Truong 2005] Truong, K. N. (2005). *Inca: an infrastructure for capture and access. supporting the generation, preservation and use of memories from everyday life*. PhD thesis, Atlanta, GA, USA. Disponível em: <http://khaitruong.com/publications/Dissertation-2005.pdf>. Acesso em: 10/08/2010.

- [Truong et al. 1999] Truong, K. N., Abowd, G. D., e Brotherton, J. A. (1999). Personalizing the capture of public experiences. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, UIST '99, pp. 121–130, Asheville, North Carolina, United States, New York, NY, USA. ACM.
- [Truong et al. 2001] Truong, K. N., Abowd, G. D., e Brotherton, J. A. (2001). Who, What, When, Where, How: Design Issues of Capture & Access Applications. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, UbiComp '01, pp. 209–224, Atlanta, Georgia, USA, London, UK, UK. Springer-Verlag.
- [Truong e Hayes 2009] Truong, K. N. e Hayes, G. R. (2009). *Ubiquitous Computing for Capture and Access*. Now Publishers Inc., Hanover, MA, USA.
- [Wang e Wu 2011] Wang, S.-L. e Wu, C.-Y. (2011). Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. *Expert Systems with Applications*, 38(9):10831 – 10838.
- [Wazlawick 2009] Wazlawick, R. S. (2009). *Metodologia da Pesquisa para Ciência da Computação*. Elsevier, Rio de Janeiro, RJ, BR.
- [Weiser 1991] Weiser, M. (1991). The computer for the 21st Century. *Scientific American*, 265(3):94–104.
- [Weiser 1993] Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Commun. ACM*, 36(7):75–84.
- [Weiser 1994] Weiser, M. (1994). The world is not a desktop. *interactions*, 1(1):7–8.
- [Weiser e Brown 1996] Weiser, M. e Brown, J. S. (1996). Beyond calculation. chapter The coming age of calm technolgy, pp. 75–85. Copernicus, New York, NY, USA.
- [White et al. 2003] White, D. R., Camacho-Guerrero, J. A., Truong, K. N., Abowd, G. D., Morrier, M. J., Vekaria, P. C., e Gromala, D. (2003). Mobile capture and access for assessing language and social development in children with autism. In *Extended Abstracts of Ubicomp 2003: The International Conference on Ubiquitous Computing*, pp. 137–140.
- [Whittaker et al. 1994] Whittaker, S., Hyland, P., e Wiley, M. (1994). FILOCHAT: handwritten notes provide access to recorded conversations. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pp. 271–277, Boston, Massachusetts, United States, New York, NY, USA. ACM.
- [Winer e Cooperstock 2002] Winer, L. R. e Cooperstock, J. (2002). The “intelligent classroom”: changing teaching and learning with an evolving technological environment. *Computer Education*, 38(1-3):253–266.
- [Wu et al. 2007] Wu, C.-I., James Teng, C.-M., Chen, Y.-C., Lin, T.-Y., Chu, H.-H., e Hsu, J. Y.-J. (2007). Point-of-capture archiving and editing of personal experiences from a mobile device. *Personal Ubiquitous Comput.*, 11(4):235–249.
- [Wu et al. 2008] Wu, S., Chang, A., Chang, M., Liu, T.-C., e Heh, J.-S. (2008). Identifying Personalized Context-Aware Knowledge Structure for Individual User in Ubiquitous Learning Environment. In *5th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (WMUTE 2008)*, pp. 95–99.

- [Xhafa et al. 2008] Xhafa, F., Barolli, L., Fernandez, R., Daradoumis, T., Caballe, S., e Kolici, V. (2008). Extending JXTA Protocols for P2P File Sharing Systems. In *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008)*, pp. 35–40.
- [Xie et al. 2010] Xie, W., Zhang, Z., Lu, C., Wang, Y., e Shen, R. (2010). PPClass - a classroom lecture broadcast platform based on P2P streaming technology. In *4th International Conference on Distance Learning and Education (ICDLE 2010)*, pp. 199–203.
- [Xu et al. 2006] Xu, W., Chen, Y., Sundaram, H., e Rikakis, T. (2006). Multimodal archiving, real-time annotation and information visualization in a biofeedback system for stroke patient rehabilitation. In *Proceedings of the 3rd ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, CARPE '06, pp. 3–12, Santa Barbara, California, USA, New York, NY, USA. ACM.
- [Zeng et al. 2010] Zeng, L., Zuo, M., e Lu, X. (2010). Research into Application of Interactive Whiteboard to Interactive Educational Mode. In *International Conference on Computational Intelligence and Software Engineering (CiSE 2010)*, pp. 1–4.
- [Zhang et al. 2008] Zhang, C., Rui, Y., Crawford, J., e He, L.-W. (2008). An automated end-to-end lecture capture and broadcasting system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4:1–23.
- [Zhong-xian et al. 2010] Zhong-xian, W., Jiang, Z., e Wen-nai, W. (2010). The Protocol Design and Implementation of a JXTA Based P2P-SIP System. In *International Conference on Internet Technology and Applications*, pp. 1–4.
- [Zhuhadar et al. 2009] Zhuhadar, L., Romero, E., e Wyatt, R. (2009). The Effectiveness of Personalization in Delivering E-learning Classes. In *Proceedings of the 2nd International Conferences on Advances in Computer-Human Interactions*, ACHI '09, pp. 130–135, Washington, DC, USA. IEEE Computer Society.

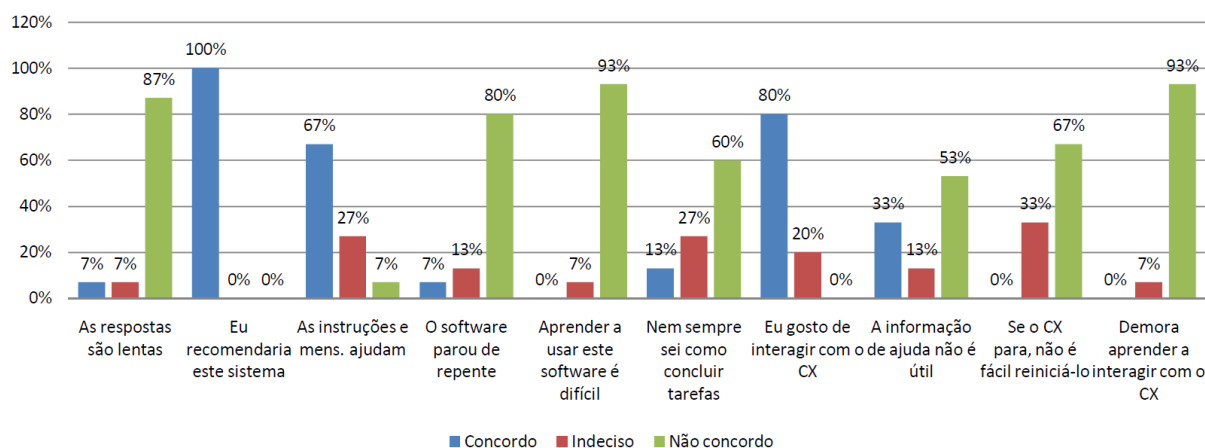
Diagrama MER do CX



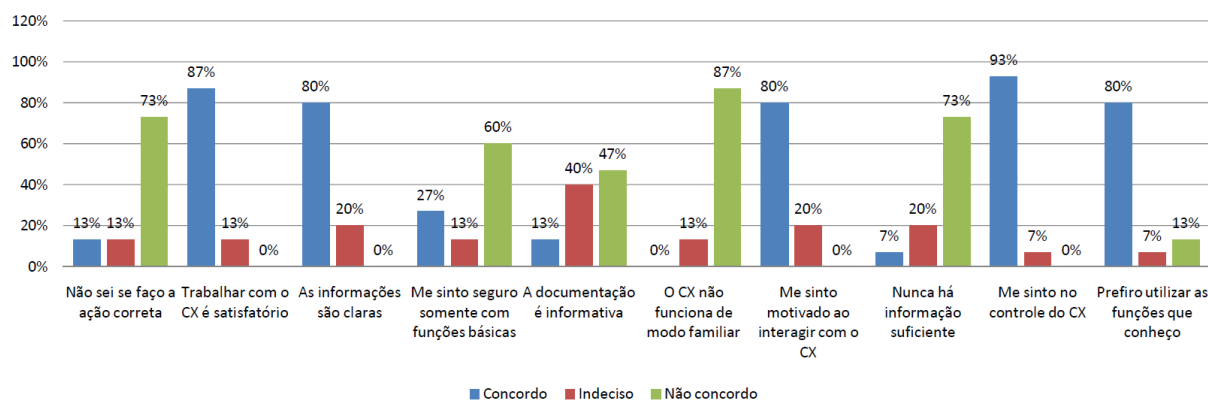
Apêndice B

Gráficos dos Resultados dos Questionários SUMI

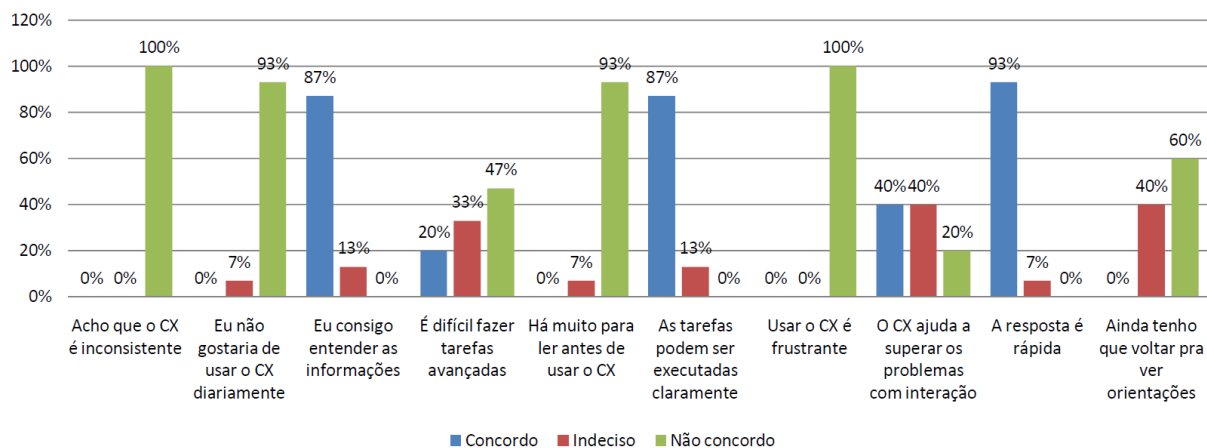
Resultados da avaliação de questionários 1/5



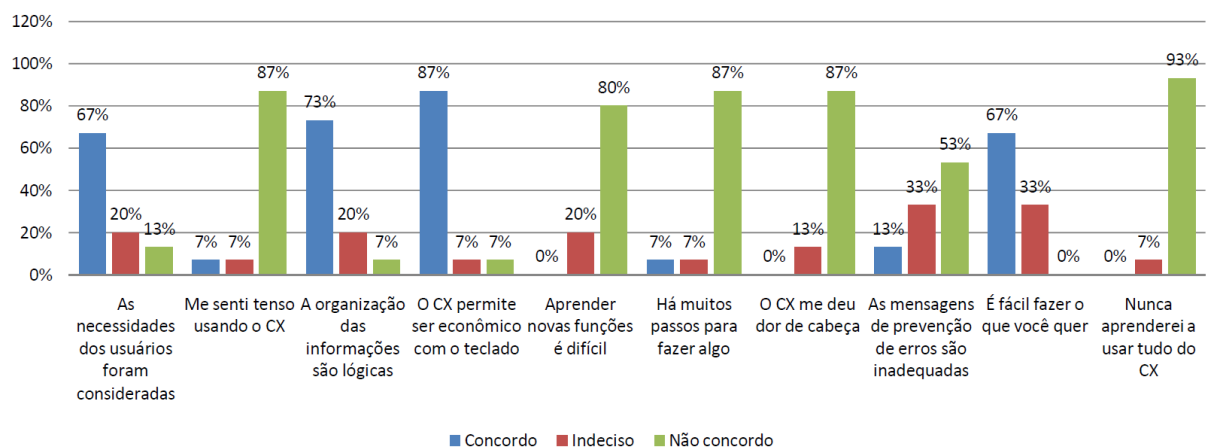
Resultados da avaliação de questionários 2/5



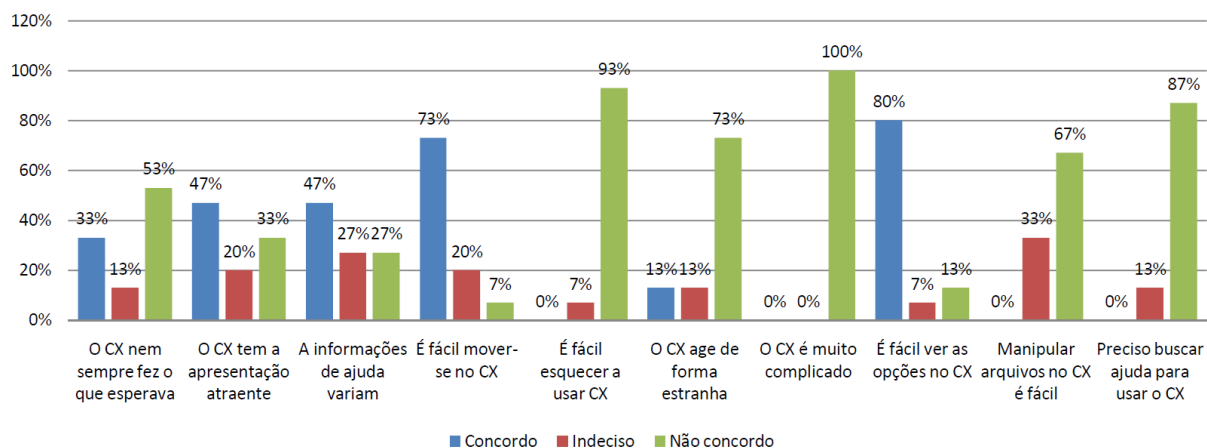
Resultados da avaliação de questionários 3/5



Resultados da avaliação de questionários 4/5



Resultados da avaliação de questionários 5/5



Anexo A

Questionário SUMI

SUMI: Questionário para avaliação de usabilidade em software

Nome:

Software: *Classroom eXperience*

Data: / / 2012

As informações aqui fornecidas são confidenciais.

Esta Avaliação leva cerca de 5 minutos para ser feita.

Você deve marcar o primeiro quadro se você concorda em geral com a sentença. Marque o quadro central se você está indeciso, não consegue se decidir ou a sentença não é relevante para o sistema, ou não se aplica. Marque o quadro direito se você discorda de modo geral com a sentença. Ao marcar o quadro da esquerda ou direita você não está necessariamente indicando que concorda ou discorda fortemente, mas sua percepção geral na maior parte do tempo.

Marque os quadros com um X.

		Concordo	Indeciso	Não Con- cordo
1	Este software responde muito lentamente à entrada de dados			
2	Eu recomendaria este software aos meus colegas			
3	As instruções e mensagens ajudam			
4	O software alguma vez parou de repente			
5	Aprender a usar este software inicialmente é difícil			
6	Nem sempre sei como concluir uma tarefa			
7	Eu gosto de interagir com este software			
8	A informação de ajuda não é muito útil			
9	Se o software para, não é fácil reiniciá-lo			
10	Demora muito para aprender a interagir com o <i>Classroom eXperience</i> (CX)			
11	Nem sempre sei se estou fazendo a ação correta			
12	Trabalhar com este software é satisfatório			
13	A informação que o CX apresenta é clara e compreensível			
14	Me sinto seguro somente usando funções básicas			

15	A documentação do software é muito informativa			
16	O CX não funciona de modo familiar para mim			
17	Me sinto motivado ao interagir com o CX			
18	Nunca há informação suficiente na tela quando necessário			
19	Me sinto no controle do CX quando estou usando-o			
20	Eu prefiro utilizar as funções que conheço melhor			
21	Eu acho que o CX é inconsistente			
22	Eu não gostaria de usar o CX diariamente			
23	Eu consigo entender a informação fornecida pelo CX			
24	É complicado fazer tarefas avançadas no CX			
25	Há muito para ler antes de poder usar o CX			
26	As tarefas podem ser executadas de modo claro usando o CX			
27	Usar o CX é frustrante			
28	O CX ajuda a superar os problemas que tive durante a interação			
29	A velocidade de resposta é rápida o suficiente			
30	Eu ainda tenho que voltar a ver as orientações			
31	Está claro que as necessidades dos usuários foram integralmente levadas em consideração			
32	Houve momentos que usando o CX me senti um pouco tenso			
33	A organização dos menus e listas de informação parecem lógicas			
34	O CX permite ao usuário ser econômico com o teclado			
35	Aprender a usar novas funções é difícil			
36	Ha muitos passos necessários para fazer algo funcionar			
37	Eu acho que o CX me deu dor de cabeça alguma vez			
38	As mensagens de prevenção de erros são inadequadas			
39	É fácil fazer exatamente o que você quer no CX			
40	Nunca aprenderei a usar tudo o que o CX oferece			
41	O CX nem sempre fez o que eu esperava			
42	O CX tem uma apresentação bem atraente			
43	Tanto a quantidade como a qualidade da informação de ajuda varia no CX			
44	É relativamente fácil mover-se de uma parte da tarefa para outra			
45	É fácil esquecer como fazer coisas no CX			
46	O CX às vezes se comporta de maneira incompreensível			
47	O CX é realmente muito complicado			
48	É fácil ver rapidamente quais são as opções em cada passo			
49	Colocar e retirar arquivos de dados no CX não é fácil			
50	Tenho que buscar ajuda a maioria das vezes quando uso o CX			

Cheque se todos os itens foram preenchidos, por favor.