Ambientes para Educacao Baseada na Web e o Modelo IMS/EDUCOM

Sergio Crespo♦, Marcus Felipe M. C. da Fontoura e Carlos José P. de Lucena
Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225, 22453-900 Rio de Janeiro, Brasil
[crespo, mafe, lucena]@inf.puc-rio.br
PUC-RioInf.MCC31/98 Agosto, 1998

RESUMO

Este artigo propõe um modelo conceitual para ambientes de educação baseados na Web (WBE), que foi desenvolvido através do estudo de vários ambientes existentes. Esse modelo conceitual é então utilizado como base de uma comparação de conceitos e funcionalidade entre esses ambientes. O modelo conceitual proposto é compatível com a plataforma EDUCOM/IMS permitindo assim que os ambientes gerados tenham interoperabilidade de conteúdo. A implementação do modelo é feita através de um framework oreientado a objetos e linguagens de domínio.

PALAVRAS CHAVES: ambientes para educação baseados na Web (WBE), modelo conceitual para WBE, plataforma EDUCOM/IMS, framework orientado a objetos, linguagens de domínio.

ABSTRACT

This paper proposes a conceptual model for Web-based education environments (WBE), which was developed through a case study of vaious existent environments. This conceptual model is used as a basis for the comparison of the concepts and functionaly of the existent environments. The proposed conceptual model is compatible with the EDUCCOM/IMS platform, allowing content interoperability for the generated environments. The model implementation is made through an object-oriented framework and a domain specific language.
KEY-WORDS: Web-based education environments, conceptual model for WBE, EDUCCOM/IMS platform, object-oriented framework, domain specific language.

♦ Professor do Departamento de Informática na Universidade do Vale do Rio dos Sinos - Unisinos-RS.

1. INTRODUÇÃO

A educação/treinamento baseada na Web (WBE) usa a WWW como meio para a publicação do material didático, aplicação de tutoriais, aplicação de provas e testes e comunicação com estudantes. Também compreende o processo de uso da Web como veículo de comunicação para a apresentação de aulas à distância (conferência multimídia).

Um processo de desenvolvimento da aprendizagem, em qualquer nível de instrução e treinamento, incorpora tipicamente as seguintes ações [8, 9]:

- Estabelecimento dos objetivos da aprendizagem;
- Localização e revisão (ou criação) de material instrucional (ex: instrumentos de diagnóstico, livros texto, software para aprendizagem (courseware), testes etc);
- Avaliação do nível de conhecimento dos alunos;
- Atribuição de material apropriado aos estudantes;
- Definição da forma de acesso dos estudantes a componentes/módulos;
- Revisão e acompanhamento do progresso dos estudantes e gerência das intervenções necessárias;
- Provisão e gestão da comunicação estudante-professor e estudante-estudante (tanto síncrona quanto assíncrona);
- Avaliação da aprendizagem;
- Relatório dos resultados da aprendizagem.

As tecnologias de educação/treinamento baseada na Web estão em pleno uso. Por exemplo, atualmente (maio de 1998) mais de 100 autores desenvolvem cursos usando o AulaNet™ (provenientes de 21 instituições) e 316 alunos estão matriculados nos 39 cursos oferecidos neste semestre; o Virtual-U e Web-CT também contam com uma grande comunidade de usuários e o mesmo vale para os demais ambientes estudados neste artigo.

No entanto, ainda existem problemas quanto a criação de conteúdo, sua disponibilização e forma de utilização. Este problema é gerado por ainda não existir uma plataforma comum para o aprendizado na Internet. Este tipo de estrutura é proposto pelo projeto EDUCOM/IMS [11], que é um investimento cooperativo de organizações acadêmicas, comerciais e governamentais dedicadas a facilitar o crescimento e viabilizar o aprendizado distribuído na Internet. O objetivo principal do EDUCOM/IMS é propor uma arquitetura aberta para o aprendizado na Web, especificando uma série de requisitos técnicos para a criação de material e ambientes de qualidade.

Um modelo conceitual foi elicitado a partir de vários ambientes estudados [2] e é utilizado neste artigo como base para uma comparação entre eles. Este modelo conceitual é implementado através de um framework orientado a objetos [12, 17] e linguagens de domínio [6] e possibilita o desenvolvimento de novos ambientes para WBE compatíveis com a plataforma EDUCOM/IMS.

2. UM MODELO CONCEITUAL PARA AMBIENTES DE EDUCAÇÃO NA WEB

Através de uma análise de domínio baseada em vários ambientes existentes definimos um modelo conceitual [2] baseado em componentes [12]. A Figura 1 ilustra esse modelo conceitual e cada um dos componentes é explicado em seguida. Os componentes funcionam como pontos de flexibilização [16, 18] do ambiente final. Desta forma a existência de cada um desses componentes em um ambiente significa que tal ambiente implementa a funcionalidade especificada pelo componente.

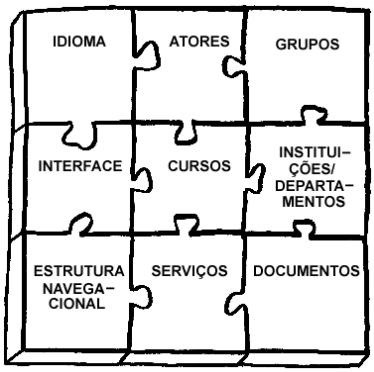

Figura 1. Modelo conceitual baseado em componentes

- Cursos: permite a criação e manutenção de cursos. O conteúdo de cada curso é controlado pelos componentes serviços e documentos;
- Atores: são as pessoas que interagem com o ambiente. Atores podem ter várias atribuições diferentes, como por exemplo alunos, professores e secretaria;
- Serviços: provêm a funcionalidade necessária para o curso. Serviços podem ser de vários tipos: serviços administrativos (agenda do curso, quadro de avisos, etc), serviços de comunicação (chat, e-mail, etc), serviços didáticos (transparências, referências na Web, etc) e serviços de avaliação (provas, auto-avaliação, etc);
- Documentos: são os artefatos maipulados pelos serviços. Por exemplo, o serviço notícias do curso necessita de um documento que possua a informação de qual é a notícia a ser exibida;
- Grupos: possibilidade de se definir grupos, possibilitando trabalho cooperativo.
- Instituições e Departamentos: capacidade de definir e customizar o ambiente para diversas instituições e departamentos;
- Idiomas: suporte a autoria e consumo de cursos em vários idiomas;
- Interface: capacidade de customização da interface do ambiente;
- Estrutura Navegacional: capacidade de customização da estrutura navegacional do ambiente.

3. COMPARAÇÃO ENTRE ALGUNS AMBIENTES

Nesta seção iremos comparar seis ambientes para WBE com base no modelo conceitual apresentado. Uma lista dos serviços implementados por cada um desses ambientes também é apresentada.

3.1 WCB

Web Course in a Box (WCB) [23] é uma ferramenta desenvolvida pela Virginia Commonwealth University para criação e mautenção de cursos na Web. Este ambiente permite a criação de páginas WWW para vários serviços tais como ementa de curso, agenda e home-page pessoais, além de funções interativas como forums de discussão e exercícios auto-corrigíveis. A autoria e o consumo do curso são ambos feitos através de browsers Web e não requerem conhecimento técnicos aprofundados.

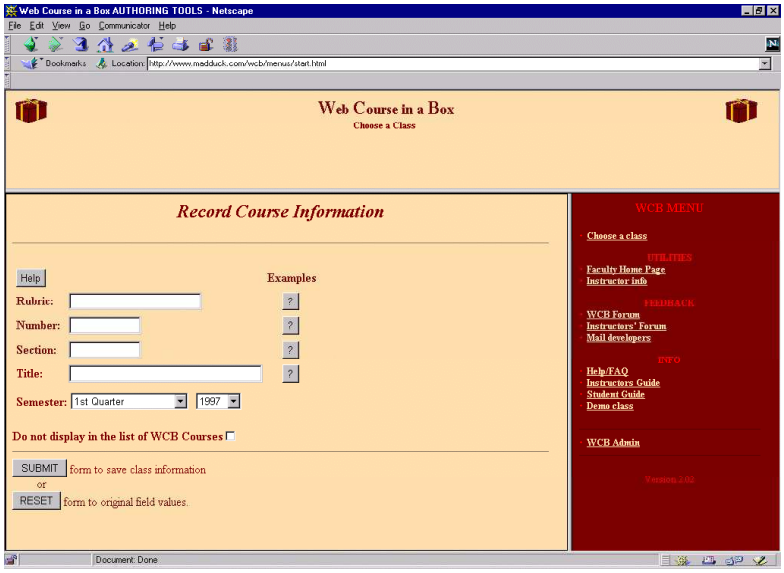

Figura 2. O ambiente WCB

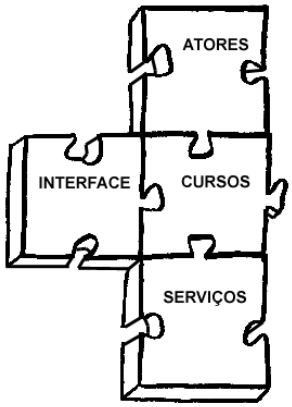

Figura 3. Modelo de componentes do WCB

O WCB não implementa o suporte a múltiplos idiomas, a capacidade de customização da estrutura navegacional, o suporte ao trabalho cooperativo, customização por intituição/departamento e a definição dos documentos. A lista de serviços implementados por esse ambiente é apresentada a seguir.

Tipos de Serviços	Serviço
Comunicação	Contato com professor Chat
Administrativos	Notícias do curso Agenda
Didáticos	Transparências Referências na Web
Gerais	Home-page de alunos Home-page de instrutores WCB Forum

Tabela 1. Serviços do WCB

3.2 Web-CT

WebCT [22] é uma ferramenta que permite a criação de ambientes de aprendizado na Web. Ela requer pouco conhecimento técnico, tanto por parte do desenvolvedor do material educacional quanto por parte do aluno.

O desenvolvedor do curso é o responsável por prover o seu conteúdo. A interatividade, estrutura navegacional e as ferramentas educacionais são fornecidas pelo ambiente, que também permite a incorporação de novas ferramentas e a alteração do layout do curso. WebCT foi desenvolvido pela University of British Columbia para facilitar a criação de diversos cursos internos [5].

Figura 4. Curso exemplo no ambiente Web-CT

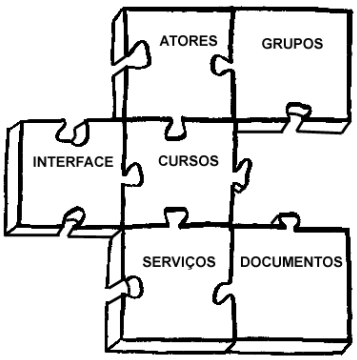

Figura 5. Modelo de componentes do Web-CT

O Web-CT não implementa o suporte a múltiplos idiomas, a capacidade de customização da estrutura navegacional e a customização por intituição/departamento. A lista de serviços implementados por esse ambiente é apresentada a seguir.

Tipos de Serviços	Serviço
Comunicação	Correio eletrônico News Chat
Administrativos	Quadro de avisos Divulgação de notas Controle do progresso do aluno Acompanhamento do curso
Avaliação	Testes periódicos

Didáticos	Glossário indexado Facilidade para anotações de aluno Material de referência do curso Quadro branco compartilhado e interativo
Gerais	Área de apresentação de alunos Arquivos de imagem indexados Indexação e busca automática

Tabela 2. Serviços do WCB

3.3 LearningSpace

A Lotus Education e a IBM são responsáves pela pesquisa e desenvolvimento do Lotus LearningSpace [13], um ambiente para desenvolvimento de serviços de apoio à educação a distância. O LearningSpace é desenvolvido sobre o ambiente de groupware Lotus Notes.

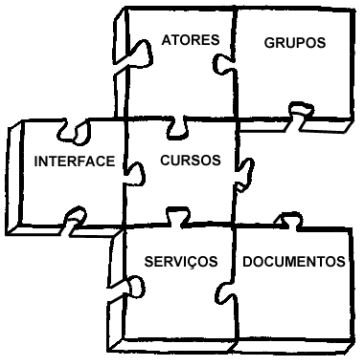

Figura 4. Modelo de componentes do LearningSpace

O LearningSpace não implementa o suporte a múltiplos idiomas, a capacidade de customização da estrutura navegacional e a customização por intituição/departamento. A lista de serviços implementados por esse ambiente é apresentada a seguir.

Tipos de Serviços	Serviço
Comunicação	Correio eletrônico News
Administrativos	Agenda
Avaliação	Exercícios Tarefas
Didáticos(Media Center)	Documentos multimídia Imagens, som e vídeo
General Services	Perfis de aluno e professor Gerente de avaliação

Tabela 3. Serviços do LearningSpace

3.4 Virtual-U

Virtual-U [19] é um ambiente baseado na Web, desenvolvido pela Simon Fraser University, que permite a integração de ferramentas e templates para a criação, manutenção e consumo de cursos on-line. O Virtual-U possui as seguintes funcionalidades:

- Criação de páginas Web de cursos;
- Estruturação de discussões interativas e atividades cooperativas entre alunos, professores e colaboradores externos;
- Criação de recursos compartilhados para disseminação do conhecimento;

- Manutenção e evolução de cursos.

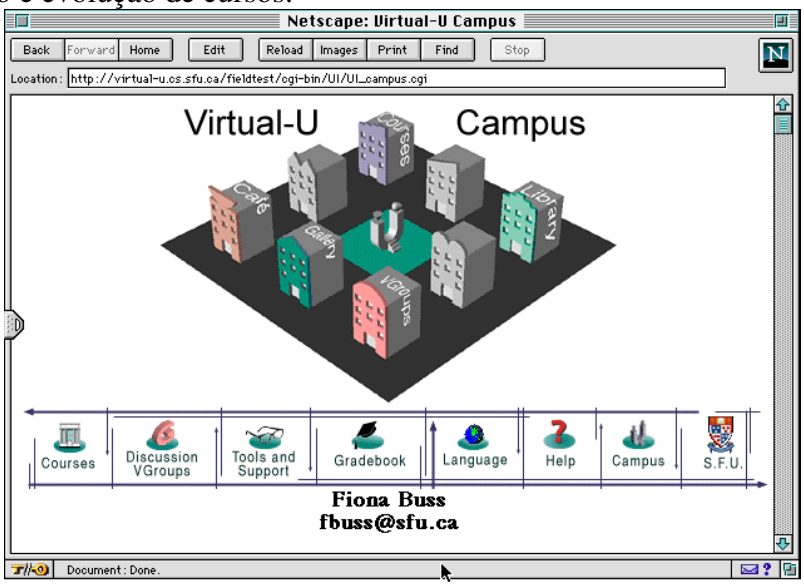

Figura 6. Ambiente Virtual-U

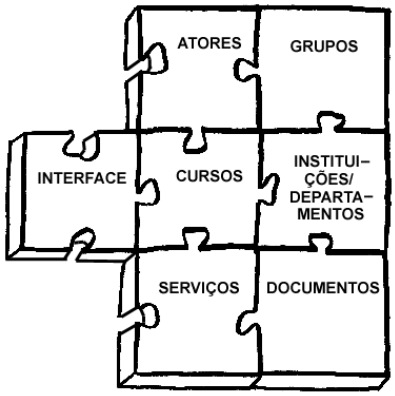

Figura 7. Modelo de componentes do Virtual-U

O Virtual-U não implementa o suporte a múltiplos idiomas e a capacidade de customização da estrutura navegacional. A lista de serviços implementados por esse ambiente é apresentada a seguir.

Tipos de Serviços	Serviço
Comunicação	Correio eletrônico News Debates
Administrativos	Agenda
Avaliação	Exercícios Tarefas
Didáticos	Seminários moderados por alunos Times de projetos Estabelecimento de metas Conferência em tempo real Chats 3D com troca de mensagem em tempo real
Gerais	Estatísticas

Tabela 4. Serviços do Virtual-U

3.5 LiveBOOKS

O LiveBOOKs [3] é um ambiente distribuído para autoria e consumo de material educacional através da Web. O ambiente foi desenvolvido pelo Computer Systems Group, da University of Waterloo, com o intuito de facilitar a criação e manutenção de cursos.

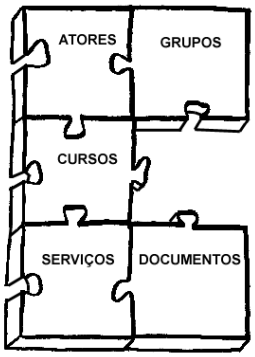

Figura 6. Modelo de componentes do LiveBOOKs

O LiveBOOKs não implementa o suporte a múltiplos idiomas, a capacidade de customização da interface e da estrutura navegacional e a customização por intituição/departamento. A lista de serviços implementados por esse ambiente é apresentada na tabela 5.

Tipos de Serviços	Serviço
Comunicação	Contato com professor Contato com tutor Contato com monitor
Avaliação	Auto-avaliação
Didáticos	LiveBOOK Notas de rodapé Biblioteca eletrônica

Tabela 5. Serviços do LiveBOOKs

3.6 AulaNet™

AulaNet™ [15] (http://les.inf.puc-rio.br/aulanet) é um ambiente de software baseado na Web, desenvolvido no Laboratório de Engenharia de Software—LES—do Departamento de Informática da PUC-Rio, para criação e assitência de cursos à distância.

Concebido a partir da experiência vivenciada com a aplicação de três cursos durante o segundo semestre de 1997 [20, 10, 21], AulaNet™ se apóia nas seguintes premissas básicas:

1. Os cursos criados devem possuir grande capacidade de interatividade, de forma a atrair a participação intensa do aluno no processo de aprendizado ("learningware").
2. O autor do curso não precisa ser necessariamente um especialista em Internet.
3. Os recursos oferecidos para a criação de cursos devem corresponder aos de uma sala de aula convencional, acrescidos de outros normalmente disponíveis no ambiente Web.
4. Deve ser possível a reutilização de conteúdos já existentes em mídia digital, através, por exemplo, da importação de arquivos.

O atendimento das premissas acima levou à formulação do conceito ***de Processo de Desenvolvimento do Aprendizado—***PDA, que exige a especificação prévia, por parte do Autor, dos recursos didáticos que ele usará durante o desenvolvimento do curso.

A utilização de AulaNet™ possibilita a criação de cursos a distância através da Internet com bastante facilidade. Os cursos são dotados de elevado grau de interatividade e possibilitam intensa participação do aluno, sem requerer do autor um conhecimento aprofundado do ambiente Web. E mais, que o curso criado pode utilizar conteúdos já existentes, gravados em mídia digital.

Atualmente existem mais de 100 cursos sendo desenvolvidos por professores de várias áreas diferentes na PUC-Rio e em outras universidades brasileiras.

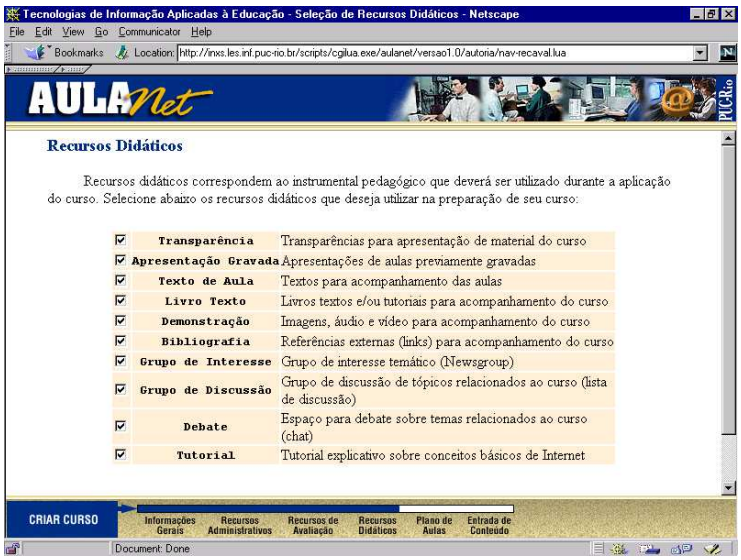

Figura 8. Ambiente AulaNet

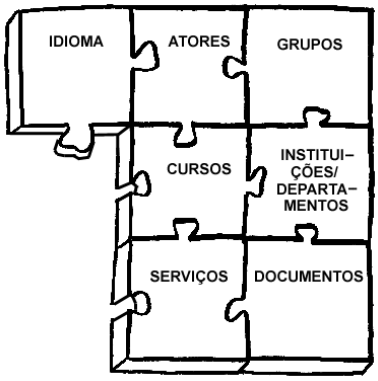

Figura 9. Modelo de componentes do AulaNet

O AulaNet^{TM} não implementa a capacidade de customização da interface e da estrutura navegacional. A lista de serviços implementados por esse ambiente é apresentada na Tabela 6.

Tipos de Serviços	Serviço
Comunicação	Grupo de interesse Grupo de discussão Contato com professor Debate
Administrativos	Agenda Notícias do curso Cadastro de instrutores
Avaliação	Teste Resultado de teste Projeto Resultado de projeto Exercício Resultado de Exercício

Didáticos	Plano de aulas Transparências Apresentação gravada Texto de aula Livro texto Demonstrações Bibliografia
Gerais	Tutorial sobre Internet Home-page de alunos Busca

Tabela 6. Serviços do AulaNet

4. O FRAMEWORK CONCEITUAL EDUCOM/IMS

O uso de computadores na educação não é uma novidade, porém com o advento das redes de computadores e em especial a Internet este uso aumentou consideravelmente, permitindo a realização de atividades com um enfoque mais colaborativo e proporcionando outras formas de interação.

O potencial da Internet tem sido bem reconhecido por escolas, universidades, organismos governamentais e organizações comerciais interessadas em promover e criar ambientes de aprendizagem. As redes de computadores, juntamente com a Internet oferecem a infra-estrutura básica para que se possa realizar ensino baseado em WEB. A interactividade na WEB em muitos casos resume-se a um clicar de mouse ou ao preenchimento de um simples formulário eletrônico. Novas tecnologias tem surgido para atuar em cima da falta de dinamicidade que o HTML não oferece. Estas tecnologias atualmente são: Java, ActiveX, JavaBeans, JavaScript, Dynamic HTML, Shockwave, Flash, RealAudio/Player, dentre outras.

Existem três tipos principais de problemas para o oferecimento de materiais on-line e para a criação de ambientes de aprendizagem:

- Deficiência de suporte para a natureza dinâmica e colaborativa da aprendizagem;
- Carência de incentivos e estrutura para o desenvolvimento e compartilhamento de conteúdos educacionais;
- Pobreza no suporte a localização e operação de materiais independentes de uma plataforma.

O projeto IMS (Instructional Management Systems – http://www.imsproject.org) é um investimento cooperativo de organizações acadêmicas, comerciais e gevernamentais dedicadas a facilitar o crescimento e viabilizar o aprendizado distribuido via Internet. O objetivo principal do IMS é propor uma arquitetura aberta para o aprendizado na Web, especificando uma série de requisitos técnicos para a criação de materiais e ambientes de aprendizagem com qualidade. Com esta especificação, o IMS oferece um framework comum para que se possa gerar e alavancar informações para o processo de aprendizagem. Um dos principais fenômenos identificados em ambientes de aprendizagem é a dinamicidade de papéis que os usuários interpretam em um dado momento.

Com a definição da plataforma IMS pretende-se atingir:

- Baixos custos para o desenvolvimento de “learningware”;
- Aperfeiçoamento da qualidade dos materiais e ambiente de aprendizagem;
- Maior acesso a aprendizagem;
- Maior customização e flexibilização das experiências com aprendizagem.

Existem vários atores dentro desta proposta, diferindo apenas quanto a seus papéis frente a esta arquitetura. Dentre os atores podemos citar:

- Aprendizes: pessoas de várias idades, habilidades, interesses e motivações;

- Professores: pessoas que transmitem seu conhecimento utilizando-se de várias metáforas, podendo ou não ser afiliadas a algum instituto de pesquisa, escola, universidade ou órgão governamental, podendo ensinar de forma individual ou participar de algum grupo de ensino;
- Coordenadores: são entidades acadêmicas, privadas, governamentais ou comunitárias que proveem crédito para iniciativas envolvendo aprendizagem, podendo participar de forma a oferecer serviços para a gerência de cursos e currículos;
- Provedores: oferecem conteúdo e serviços oriundos de trabalhos de um único autor, de um grupo ou de outros provedores agregados tais como revistas, jornais, etc.

O IMS em sua arquitetura oferece ferramentas para que se possa ter facilidade de gerenciamento destas partes, produzindo desta maneira materiais educacionais de alta qualidade e reusáveis. Muitas escolas e universidades já tem investido milhões de dolares em infra-estrutura para que se possa criar ambientes de aprendizagem. A plataforma IMS pretende gerar condições para que toda a infra-estrutura já criada possa ser utilizada nas produção de ambientes de aprendizagem.

O modelo de sistemas do IMS utiliza um alto nível de organização de serviço definindo:

- *Meta-Dados*: são campos e valores associados que descrevem um recurso físico ou eletrônico. IMS tem definido uma extensa coleção de meta-dados para recursos educacionais, recursos de treinamento, conteúdo, informação pessoal e informação certificada, dentre outros;
- *Dicionarios e Tipos*: para o armazenamento de meta-dados, IMS define um dicionário e uma coleção de termos recomendados para a classificação dos diferentes tipos de recursos de aprendizagem;

Site de Informação de Meta-dados: o IMS desenvolveu em conjunto com o NIST (National Institute os Standards and Technology), um repositório de meta-dados de termos que deverão ser um recurso para uma grande comunidade de desenvolvedores de conteúdo, usuários, e provedores de serviços, facilitando a identificação e o reuso de meta-dados de informações comuns (http://sdct-sunsrv1.ncsl.nist.gov/~boland/ims.html). Lá pode-se encontrar também uma coleção adicional de termos que são especializações do núcleo do modelo IMS estabelecidos por indústrias e grupos pertencentes a determinados domínios do conhecimento.

O IMS focaliza sua atenção nos seguintes tópicos:

- Interoperabilidade: os vários ambientes de aprendizagem poderão se comunicar e interagir mutuamente, de forma análoga a um grupo de alunos pertencentes a grupos diferentes e trabalhando em conjunto por meio de um programa de Chat;
- Customização e extensibilidade: a customização de um sistema é viável através de uma combinação de sua partes interoperáveis requerendo-se para isso que os sistemas sejam construídos de forma a beneficiar a generalização de seus módulos e que tenham procedimentos padrões frente aos já existentes no mercado;
- Colaboração: por colaboração entende-se a troca entre alunos e o ambiente de aprendizagem propriamente dito, que consiste de pessoas e recursos.

O projeto IMS procura ser extensível frente a novas tecnologias tais como:

- Objetos distribuídos: COM (Microsoft's Component Object Model) e Corba da OMG (Object Management Group). Ambos os modelos dão suporte Java e o modelo Corba dá suporte também a Java/RMI. A existência deste modelos permite uma neutralidade no que tange ao desenvolvimento de componentes apropriados para um determinado público alvo.
- Agentes para facilitar o processo de aprendizagem, atuando muitas vezes como um tutor.
- O uso da realidade virtual e de ambientes imersivos de aprendizagem devem aumentar em virtude de um aumento de banda e das capacidades gráficas dos atuais computadores

- As especificações do IMS deverão utilizar o padrão XML™ (Extensible Markup Language) como sintaxe para dados e objetos.

A figura abaixo ilustra a organização global do Sistema IMS, mostrando como o servidor IMS interage com ferramentas de autoria, máquinas de busca, servidores de conteúdo e servidores de perfis para criar um ambiente de aprendizagem personalizado para ensino e autoria.

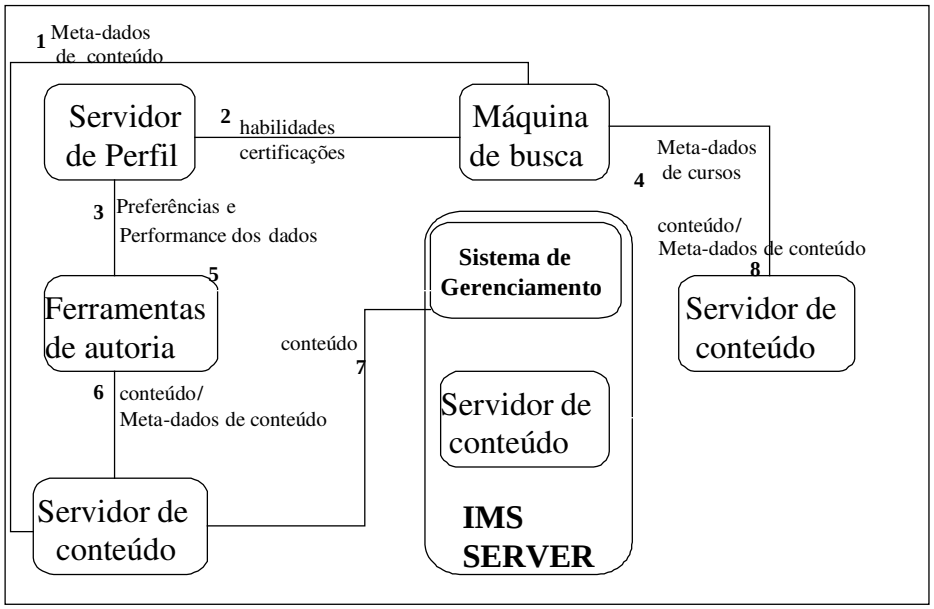

Figura 10. Modelo do sistema IMS

Onde:

1. A máquina de busca pode utilizar meta-dados de informação para buscar no servidor de conteúdo, tipos específicos de materiais de aprendizagem;
2. A máquina de busca pode buscar no servidor de perfis informações de pessoas com determinadas características;
3. Uma ferramenta de autoria pode buscar no servidor de perfis informações que permitam customizar a apresentação de forma personalizada e eficiente;
4. A máquina de busca pode obter meta-dados de cursos no servidor de conteúdo de uma organização hospedada em um servidor IMS;
5. Ferramentas de autoria podem realizar intercâmbio de conteúdos com o sistema de gerenciamento IMS;
6. Ferramentas de autoria podem interagir com servidores de conteúdo para obter ou oferecer meta-dados de conteúdo ou conteúdo propriamente dito;
7. O servidor de conteúdo pode oferecer conteúdo para o sistema de gerenciamento IMS;
8. Deverão existir vários servidores de conteúdo que pode ser localizado por máquinas de busca.

A seguir veremos a apresentação do modelo conceitual apresentado compatível com as especificações do projeto IMS.

5. IMPLEMENTAÇÃO DO MODELO CONCEITUAL

Essa seção apresenta uma breve descrição da implementação do modelo conceitual apresentado. Uma descrição detallhada é apresentada em [4]. Essa implementação, feita em CGI-Lua [7], se baseia em um framework orientado a objetos [12, 17] que implementa todos os componentes apresentados no modelo e em linguagens de domínio [6] que são utilizadas para a instanciação do framework. A estrutura de classes do framework ALADIN é apresentada a seguir usando diagramas OMT.

ALADIN permite que várias instituições usem o ambiente gerado simultâneamente. Cada instituição pode conter vários departamentos. Os cursos são relacionados com as instituições e cada curso contém vários atores. Cada ator pode ser de vários tipos (por exemplo estudantes, professores, autores). Um curso é composto por uma seleção de serviços. Cada serviço pode ser de dois tipos: interno, que é implementado pelo próprio ambiente, e externo, que utiliza uma aplicação WWW externa ao ambiente. Exemplos de serviços externos podem ser servidores de Chat, CU-SeeMe e servidores de lista. Essa estrutura de design[1] é mostrada na Figura 11.

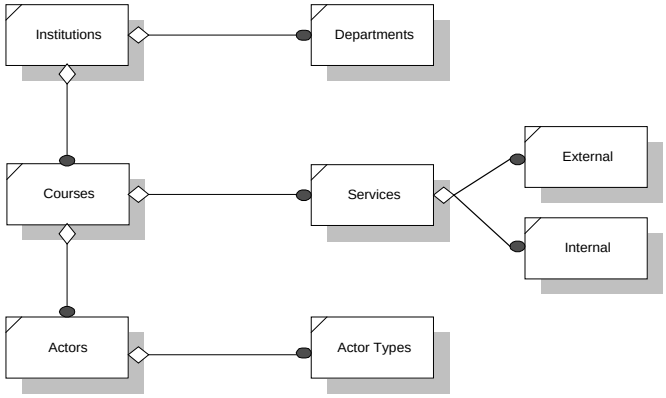

Figura 11. Estrutura de classes do framework ALADIN

ALADIN permite que o usuário final modifique a representação (interface) visual de cada entidade do ambiente gerado, como é mostrado na Figura 12.

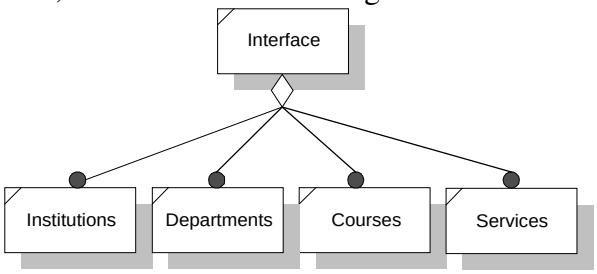

Figura 12. Estrutura de classes do componente interface

Cada ambiente para WBE é composto de um site para cada tipo de ator definido. Nos ambientes comparados na Seção 3 sempre existem dois sites, um para aprendizado (learning site), que é utilizado pelo aluno, e outro para autoria (authoring site), que é utilizado pelo autor. No entanto, o framework ALADIN permite que sejam gerados sites WWW para os outros tipos atores definidos no sistema, como por exemplo sites para monitores e secretaria. Todos os sites gerados pelo framework ALADIN podem definir várias estruturas de navegação. A Figura 13 mostra o diagrama OMT para as classes que definem a estrutura dos sites gerados.

[1] Note que não existem classes para a implementação dos componentes de grupos e documentos, que podem ser implementados pelas classes de serviços.

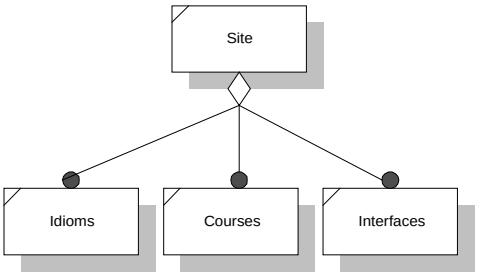

Figura 13. Estrutura de classes dos sites gerados pelo framework ALADIN

Duas linguagens de domínio (DSL) [6] são utilizadas para a instanciação do framework ALADIN.

1. Linguagem Educacional: usada na definição dos componentes educacionais (cursos, atores, serviços, grupos, documentos, instituições e departamentos);
2. Linguagem Navegacional: usada na definição do idioma, interface e estrutura navegacional.

Todos os componentes utilizados pelo novo ambiente que se deseja gerar são descritos através de programas nestas linguagens.

5.1 Linguagem Educacional

O código escrito na linguagem educacional é transformado (com o auxílio do sistema transformacional Draco-PUC [14]) em código em duas linguagens: um em Microsoft Access Basic, usado na definição da estrutura da base de dados que será utilizada pelo ambiente, e outro em CGILua, que prove os métodos necessários para o acesso à base de dados.

O trecho de programa a seguir é escrito em Linguagem Educacional.

```
Institution "PUC-RIO", "Pontificia Universidade Catolica - Rio de Janeiro", "PUC.gif";
Department "CETUC", "TELECOMUNICATIONS", "CETUC.GIF";
Department "CS", "COMPUTER SCIENCE", "CS.GIF";

Actor Type Teacher, "Teacher"
       {      name String;
              description Memo;
              Photo Image; };

Actor Type Student, "Student"
       {      name String;
              description Memo;
              period Integer;
              address String;
              average Real; };
Course
       {      name String;
              code String;
              syllabus Memo;
              description Memo;
              image Image; };
Service "CourseNews"
       {      news Memo;
              initialDate Date;
              finalDate Date; }
       read = [ Student ]
       write = [ Teacher ];
```

Note que cada operador da linguagem manipula não só as informações utilizadas pelo ambiente que será gerado, mas também a meta-informação sobre a estrutura dos dados. A definição dos atributos de um curso no operador Course é um exemplo. Essa meta-informação é usada na geração de meta-tags no padrão XML [11], conforme definido no projeto EDUCOM/IMS. Essa abordagem garante que cada ambiente gerado seja compatível com a plataforma IMS, assegurando interoperabilidade de conteúdo.

Parte[2] do código em Microsoft Access Basic gerado pela transformação do trecho de código em Linguagem Educacional acima é apresentado a seguir.

```
Dim wrkDefault As Workspace
```

[2]Esse trecho de código cria a tabela para a definição de instituições e adiciona um registro a ela.

```
Dim rst As Recordset
Set wrkDefault = DBEngine.Workspaces(0)
Dim outputDB As Database
Set outputDB = wrkDefault.CreateDatabase("ALADIN.mdb", dbLangGeneral, dbVersion30)

Dim Department As TableDef
Set Department = outputDB.CreateTableDef("Department")

Department.Fields.Append Department.CreateField("Id",dbLong)

Dim idxDepartment As Index
Set idxDepartment = Department.CreateIndex("PrimaryKey")
idxDepartment.Fields.Append idxDepartment.CreateField("Id")
idxDepartment.Primary = True
idxDepartment.Unique = True
Department.Indexes.Append idxDepartment

outputDB.TableDefs.Append Department
Dim DepartmentNameField As Field
Set DepartmentNameField = Department.CreateField("Name", dbText,255)
DepartmentNameField.AllowZeroLength = True
Department.Fields.Append DepartmentNameField
Dim DepartmentDescriptionField As Field
Set DepartmentDescriptionField = Department.CreateField("Description", dbMemo)
DepartmentDescriptionField.AllowZeroLength = True
Department.Fields.Append DepartmentDescriptionField
Dim DepartmentImageField As Field
Set DepartmentImageField = Department.CreateField("Image", dbText,255)
DepartmentImageField.AllowZeroLength = True
Department.Fields.Append DepartmentImageField
Dim Institution As TableDef
Set Institution = outputDB.CreateTableDef("Institution")

Set rst = outputDB.OpenRecordset ("Institution",dbOpenDynaset)
With rst
.AddNew
!Id = 10
 !Name = "PUC-Rio".Update
.Close
End With
```

A execução do codigo em Microsoft Access Basic gera a base de dados (em Microsoft Access) que será utilizada pelo ambiente WBE instanciado pelo framework ALADIN. Como descrito acima, a transformação do código em Linguagem Educacional também gera métodos para o acesso aos dados. As interfaces de métodos apresentadas a seguir, utilizando a sintaxe de CGILua, definem os métodos gerados para a manipulação de cursos.

```
luaTable = getCourse(name, code, syllabus, description, image)
addCourse (name, code, syllabus, description, image)
updateCourse(oldName, name,  code, syllabus, description, image)
deleteCourse(name)
```

Esses métodos encapsulam os comandos SQL, permitindo que mesmo usuários que não possuam conhecimento de como manipular bases de dados relacionais gerem ambientes WBE através do framework ALADIN.

5.2 Linguagem Navegacional

O código em Liguagem Navegacional é transformado (também com auxílio da ferramenta transformacional Draco-PUC) em arquivos HTML que definem a interface e estrutura navegacional do ambiente WBE gerado. Esses arquivos também possuem trechos de código CGILua, correspondentes aos métodos de acesso descritos acima.

Um exemplo de código em Linguagem Navegacional é apresentado a seguir.

```
Language "English"
Language "Portuguese"
Text "title1", " English", "Resources"
Text "title1", "Portuguese", "Recursos"
Image "img1", " English", "c:\ing\img.gif"
Image "img1","Portuguese","c:\port\img.gif"

a := template("c:\templates\temp1.html")
b := template("c:\templates\temp2.html")
c := template("c:\templates\temp3.html")
```

```
b.next := c
b.previous := a
```

A linguagem possui um operador para a definição dos idiomas presentes no ambiente gerado (Language). Os textos e imagens usados pelo ambiente são então definidos nos vários idiomas. Note que os arquivos HTML na verdade são templates que possuem tags especiais para a definição de textos e imagens. Neste exemplo, a tag <ALADIN-TEXT>title1</ALADIN-TEXT> seria substituida pelo texto "Resources", caso o idioma selecionado fosse Inglês, e pelo texto "Recursos", se o idioma selecionado fosse Português.

O mesmo vale para os links hipertextuais (conforme definido no componente de Estrutura Navegacional do modelo conceitual). No exemplo acima a tag <ALADIN-LINK>previous</ALADIN-LINK> no template temp2.html seria substituida pelo código HTML <A HREF="c:\templates\temp1.html">previous</A>, enquanto a tag <ALADIN-LINK>next</ALADIN-LINK> seria substituida pelo código HTML <A HREF="c:\templates\temp3.html">next</A>. Essa abordagem permite a definição de todos os links em um arquivo separado, provendo uma maior flexibilidade e legibilidade.

5.3 Exemplos de uso do framework ALADIN

Essa seção descreve como os sites de autoria e aprendizado do ambiente AulaNet™ [15] foram gerados pelo framework ALADIN. A estrutura geral do site de autoria é mostrada na Figura 14.

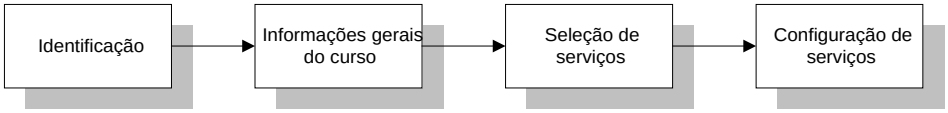

Figura 14. Estrutura global do site de autoria do ambiente AulaNet

Na identificação, o autor informa o seu userid e senha ao sistema. Durante a etapa de informações gerais do curso, toda a informação básica sobre o curso precisa ser completada, tais como nome do curso, descrição e ementa[3]. Então o autor deve informar ao sistema quais são os serviços necessários para o curso. Finalmente, na etapa Configuração de serviços, cada um dos serviços disponíveis no curso é configurado. Quando todos as etapas são completas o curso está pronto e o seu site de aprendizado é automaticamente gerado.

A estrutura e o código HTML (com rotinas CGILua) gerados pelo framework ALADIN para a etapa de seleção de serviços é apresentada a seguir.

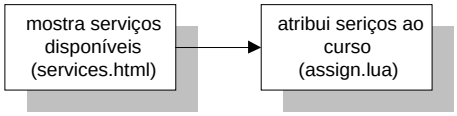

Figura 15. Estrutura detalhada da etapa de seleção de serviços

```
**************************************
*             services.html          *
**************************************
<HTML>
<HEAD>
   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
   <META NAME="Author" CONTENT="Leonardo Moura">
   <META NAME="GENERATOR" CONTENT="Mozilla/4.02 [en] (Win95; I) [Netscape]">
   <TITLE>Show Available Services</TITLE>

<BODY BGCOLOR="#FFFFFF" TEXT="#000000" topmargin=0 leftmargin=0>
<H1> Please select the services you want to use in your course </H1>

<FORM NAME="data" TARGET="_top" METHOD="post" ACTION="assign.lua">

<!--$$
-- Select all the services availabe to the current intitution.
-- The result of this selection is stored in a lua table.
```

[3] Note que a informação de quais são os campos que devem ser preenchidos na etapa de informações gerais do curso é fornecida pelo operador Course da Linguagem Educacional.

```
table = getService(currentInstitution)

while (table ~= nil) do
  write('<td valign=middle>')
  write('<INPUT TYPE="Checkbox" NAME="'..table[0]..'" VALUE="'..table[0]..'">')
  write('</td>')

  moveNext(table)
end

$$-->

<INPUT TYPE="Submit" VALUE="Next">
</FORM>

</BODY>
</HTML>

**************************************
*              assign.lua            *
**************************************
table = getService(currentInstitution)
while (table ~= nil) do
  -- test to check if the resource is select or not
  -- if true add the resource to the course
  if (cgi.table[0] = "ON") then
       has(course, table[0])
  moveNext(table)
end
```

A estrutura do site de aprendizado é apresentada na Figura 16. O aluno passa por uma página de identificação, fornecendo o seu userid e senha. Em seguida o sistema apresenta uma lista de todos os cursos em que o aluno está matriculado. O aluno deve selecionar um dos cursos, e o sistema então mostra a página de abertura do curso, que contém todas as informações gerais do curso e a lista dos serviços disponíveis. Quando o aluno seleciona o serviço desejado, um frame que implementa tal serviço é apresentado.

A estrutura e o código HTML (com rotinas CGILua) gerados pelo framework ALADIN para a etapa lista de cursos é apresentada a seguir.

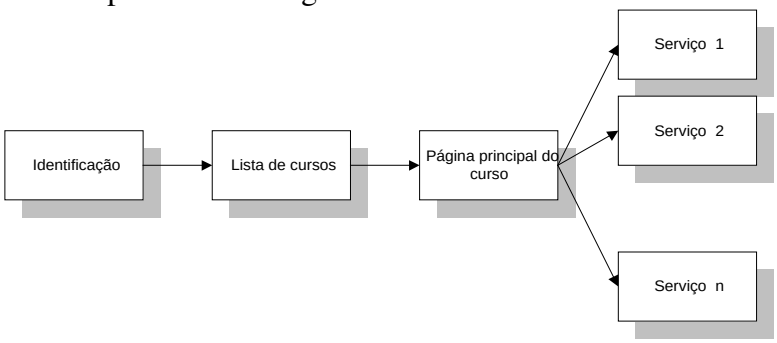

Figura 16. Estrutura global do site de aprendizado do ambiente AulaNet

```
**************************************
*            courselist.html         *
**************************************
<HTML>
<HEAD>
   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
   <META NAME="Author" CONTENT="Leonardo Moura">
   <META NAME="GENERATOR" CONTENT="Mozilla/4.02 [en] (Win95; I) [Netscape]">
   <TITLE>Show List of Cousers</TITLE>

<BODY BGCOLOR="#FFFFFF" TEXT="#000000" topmargin=0 leftmargin=0>
<H1> Please select the course you want to attend now</H1>

<FORM NAME="data" TARGET="_top" METHOD="post" ACTION="coursemain.html">

<!--$$
table = getActorsCourses(currentActor, student)

while (table ~= nil) do
  write('<td valign=middle>')
```

```
  write('<INPUT TYPE="Radio" NAME="'..table[1]..'" VALUE="'..table[1]..'">')
  write('</td>')

  moveNext(table)
end

$$-->

<INPUT TYPE="Submit" VALUE="Attend">

</FORM>
</BODY>
</HTML>
```

A semântica formal de cada um dos componentes do modelo conceitual pode ser descrita através dos operadores das linguagem de domínio do framework ALADIN [4].

6. CONCLUSÕES E TRABALHOS FUTUROS

O modelo conceitual apresentado mostrou-se adequado para a comparação dos conceitos e funcionalidades de ambientes para WBE. Essa comparação é bastante útil para o desenvolvimento de novos ambientes mais abrangentes.

Como o domínio de aplicação de ambientes para WBE ainda não é completamente entendido a necessidade de um ambiente que permita o desenvolvimento rápido de novos ambientes para WBE é uma meta desejável. O modelo conceitual aqui descrito é implementado [4] através do framework orientado a objetos ALADIN. Linguagens de domínio são usadas no processo de instanciação do framework, permitindo o desenvolvimento de novos ambientes de maneira rápida e a baixos custos por professores com a assistência de engenheiros de software. Os ambientes gerados por essa arquitetura são compatíveis com os novos padrões estabelecidos no projeto EDUCON/IMS [11] permitindo a integração com outros servidores IMS, através do uso de meta-dados para garantir a interoperabillidade dos conteúdos gerados.

O componente de estrutura navegacional existe apenas no modelo conceitual, não sendo presente em nenhum dos ambientes analisados. A razão para isso é o fato de que a nova versão do AulaNet™ está sendo gerada pelo framework ALADIN e atenderá aos requisitos impostos por este componente.

Em paralelo estamos realizando o desenvolvimento de um novo ambiente para WBE, integrando os ambientes AulaNet™ e LiveBOOKs [1].

7. REFERÊNCIAS

1. P. Alencar, D. Cowan, S. Crespo, M. F. Fontoura, and C. J. Lucena, "OwlNet: An Object-Oriented Environment for WBE", Proceedings of the Second Argentine Symposium on Object-Orientation, ASOO'98, Buenos Aires, Argentina, 1998 (to appear).

2. P. Alencar, D. Cowan, S. Crespo, M. F. Fontoura, and C. J. Lucena, "Using Viewpoints to Derive a Conceptual Model for Web-based Education", MCC17/98, Monografias em Ciência da Computação, Departamento de Informática, PUC-Rio, 1998 (also submitted to ICSE'99).

3. D. Cowan, "An Object-Oriented Framework for LiveBOOKs", Technical Report, CS-98, University of Waterloo, Ontario, Canada, 1998.

4. D. Cowan, S. Crespo, M. F. Fontoura, C. J. Lucena, and L. M. Moura, "ALADIN: An Architecture for Learningware Applications Design and Instantiation", MCC/98, Monografias em Ciência da Computação, Departamento de Informática, PUC-Rio, 1998 (also submitted to WWW Journal).

5. M. Goldeberg and S. Salari, “World Wide Web-Course Tool: An Environment for Building World Wide Web-based Courses”, Computer Networks and ISDN Systems, 28, pages 1212-1231, 1996.
6. P. Hudak, “Building Domain-Specific Embedded Languages”, Computing Surveys, 28A(4), ACM, 1996.
7. R. Iersalimschy, R. Borges, and A. M. Hester, “CGILua A Multi-Paradigmatic Tool for Creating Dynamic WWW Pages”, SBES’97 (Simpósio Brasileiro de Engenharia de Software), 1997.
8. International Data Corporation, Special Report, “Web-based Education/Training”, 1997.
9. Internet 2, <http://www.internet2.edu>.
10. ICC, <http://www.les.inf.puc-rio.br/icc> (Português).
11. IMS, <http://www.imsproject.org/specs.html>.
12. R. Johnson, “Frameworks = (Components + Patterns)”, Communications of the ACM, Volume 40, Number 10, October 1997.
13. LearningSpace, <http://www.lotus.com/home.nsf/welcome/learnspace>.
14. J. C. S. P. Leite, M. Sant’anna, and F. G. Freitas, “Draco-PUC: a Technology Assembly for Domain Oriented Software Development”; Proceedings of the 3rd IEEE International Conference of Software Reuse; 1994.
15. C. Lucena, H. Fuks, R. Milidiu, L. Macedo, N. Santos, C. Laufer, M. Ribeiro, M. Fontoura, R. Noya, S. Crespo, V. Torres, L. Daflon, and L. Lukowiecki, “AulaNet™ - An Environment for the Development and Maintenance of Courses on the Web”, in ICEE’98, 1998 (to appear).
16. W. Pree, “Design Patterns for Object-Oriented Software Development”, Addison-Wesley, 1995.
17. D. Roberts and R. Johnson, “Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks” in “Pattern Languages of Program Design 3”, Addison-Wesley, 1997.
18. H. A. Schmid, “Systematic Framework Design by Generalisation”, Communications of the ACM, Volume 40, Number 10, October 1997.
19. Simon Fraser University, <http://virtual-u.cs.sfu.ca/vuweb/>.
20. Sociedade da Informação, <http://www.les.inf.puc-rio.br/socinfo> (Português).
21. Transferência de Calor, <http://www.les.inf.puc-rio.br/transcal> (Português).
22. University of British Columbia, <http://homebrew.cs.ubc.ca/webct/>.
23. Virginia Commonwealth University, <http://views.vcu.edu/wcb/intro/wcbintro.hml>.