

Integrando a Produção de Documentos Didáticos com Ontologias: o Sistema e-Nsino

Giovani Rubert Librelotto e Marcos Luís Cassal

Abstract— This paper describes a system to provide support for E-Learning content production. Basically, LMS systems provide functionalities to manage student records, to facilitate communication between students and teacher, to control accesses and produce statistics, schedules, evaluation, and an open platform to help teachers make lecture content available online. However they do not dictate what kind of technology or format should be used to prepare those contents. Although this issue can be seen as an advantage in certain contexts it leads to a format anarchy and makes support for content production impossible. Here is where *e-Nsino* comes into the scene.

Index Terms—Knowledge representation, Learning systems, Semantic networks. e-learning.

I. INTRODUÇÃO

HOJE em dia, a Web é um dos veículos mais atraentes para a distribuição e acesso à informação. A sua aplicação na área do ensino à distância era uma questão de tempo. Desde meados da década de 90, realizam-se experiências de utilização da Web para o ensino à distância. Naquela época, utilizavam-se *mailing-lists* para interagir com os alunos e páginas Web para disponibilizar documentos.

Atualmente, a procura de cursos no formato *E-Learning* cresce diariamente. A possibilidade de poder seguir um curso a partir do conforto de casa ou simplesmente eliminando distâncias e barreiras geográficas é um conceito muito atrativo. No entanto, o *E-Learning* exige a preparação de documentos e a gestão de alunos, o que consome tempo e recursos humanos.

É aqui que surgem os sistemas de gestão de ensino (LMS – *Learning Management Systems*). Já existe um número considerável destes sistemas em utilização por todo o mundo. Alguns deles, devido à qualidade de serviços que oferecem, são considerados referências: *Blackboard*, *Luvit* e *WebCT*. Um sistema destes fornece funcionalidades para: gerenciar os processos dos alunos; facilitar e gerenciar a comunicação entre alunos, e entre estes e o professor; controlar os acessos ao sistema e produzir estatísticas; organizar horários; efetuar a avaliação dos alunos em uma plataforma aberta à publicação de documentos. No entanto, estes sistemas não fornecem

qualquer tipo de regra quanto à tecnologia ou ao formato a ser utilizado na preparação dos documentos. Apesar disto poder ser visto como uma vantagem, em alguns contextos pode levar a uma proliferação anárquica de formatos e à impossibilidade de criação de um suporte à produção de documentos.

Olhando para o estado da arte, pode-se dividir a produção de documentos em duas áreas: metadados e conteúdo propriamente dito. Até ao momento, a comunidade acadêmica tem-se preocupado com a meta-informação associada à aquilo que se convencionou designar por objetos de aprendizagem (*Learning Object*). Atualmente existem várias normas para a meta-informação de um objeto de aprendizagem como o LOM (*Learning Object Metadata*) [18], o SCORM (*Sharable Content Object Reference Model*) [1] e o *Learning Design* da IMS [10]. Neste projeto, inclui-se alguma meta-informação nos objetos de aprendizagem (um subconjunto pequeno); porém o projeto centrar-se-á no conteúdo dos documentos (quanto a este, não há normas nem propostas) e na construção de uma rede semântica envolvendo os conceitos (tópicos) abordados nos documentos didáticos.

Pensando nisso, apresenta-se neste artigo um ambiente orientado ao docente, permitindo a ele a criação de documentos referentes ao dia a dia de uma sala de aula de uma forma amigável e estruturada. Além de proporcionar editores especializados, este ambiente permite a geração automática de uma interface Web para disponibilizar os documentos. Esta interface é criada a partir da representação do conhecimento extraída dos documentos editados pelo docente.

O presente artigo está estruturado da seguinte forma: a Seção II introduz os *Topic Maps*, enquanto que a Seção III apresenta o sistema *e-Nsino*, com ênfase na estrutura deste sistema. A estruturação, edição e transformações dos documentos didáticos são abordadas na Seção IV. A Seção V descreve a extração da ontologia e a produção das páginas Web referentes aos documentos didáticos. A avaliação do ambiente *e-Nsino* está na Seção VI. Conclui-se o artigo na Seção VII com a apresentação dos trabalhos futuros.

II. REPRESENTAÇÃO DO CONHECIMENTO, MAPAS DE CONCEITOS E TOPIC MAPS

Segundo Ausubel [2], estudar e conhecer a estrutura cognitiva dos alunos e intervir na sua reorganização é um objetivo da maior importância que, como tal, tem sido alvo de numerosos estudos. Trata-se, em primeiro lugar, de procurar compreender como se processa a aprendizagem para, em

G. Librelotto e M. Cassal são professores na UNIFRA (Centro Universitário Franciscano), Rua dos Andradas, 1614, 97010-032, Santa Maria, RS, Brazil. Email: {giovani,cassal}@unifra.br
DOI (Digital Object Identifier) Pendiente

seguida, utilizar esse conhecimento como um instrumento mais eficaz de ensino-aprendizagem na sala de aula.

Novak e Gowin (1988, citado por Nunes, 2000:pág. 53) [7], seguindo a linha de pensamento de Ausubel, afirmam que existem dois tipos de aprendizagem: aprendizagem memorística e aprendizagem significativa. Para estes autores a aprendizagem memorística corresponde à maior parte da aprendizagem escolar, sendo caracterizada por não haver nenhum compromisso afetivo para relacionar os novos acontecimentos com a aprendizagem prévia e a aprendizagem não está relacionada com as experiências, com acontecimentos ou objetos.

Por outro lado, a aprendizagem significativa tem, segundo os mesmos autores, as seguintes características, completamente opostas:

- a incorporação dos novos conhecimentos na estrutura cognitiva faz-se de modo não arbitrário, substantivo e não literal (ou seja, com fortes ligações ao conhecimento prévio do indivíduo);
- há um esforço deliberado para ligar os novos conhecimentos a conceitos de nível superior, mais inclusivos, na estrutura cognitiva;
- a aprendizagem é relacionada com as experiências e com acontecimentos ou objetos;
- há um compromisso afetivo para relacionar os novos conhecimentos com a aprendizagem prévia.

De entre as numerosas técnicas, utilizadas tanto na identificação e caracterização das estruturas conceituais individuais, como na promoção da sua reorganização destaca-se a elaboração de mapas de conceitos. Os Mapas de Conceitos, usados conjuntamente com outras estratégias de ensino, poderão facilitar a compreensão da natureza do conhecimento científico, quer realçando o seu carácter evolutivo, quer ilustrando a forma como esse conhecimento é construído.

Novak [8] define Mapa de Conceitos como um instrumento para organizar e representar conhecimento. O mapa de conceitos é uma representação gráfica em duas dimensões de um conjunto de conceitos, construída de tal forma que as relações entre eles sejam evidentes. Os conceitos aparecem dentro de caixas, nos nós do grafo, enquanto que as relações entre os conceitos são especificadas através das frases de ligação, nos arcos que unem os conceitos. A dois ou mais conceitos conectados por frases de ligação criando uma unidade semântica, chamada proposição. De acordo com Novak, o eixo vertical expressa um modelo hierárquico para os conceitos onde os mais gerais aparecem na parte superior e os mais específicos nas partes inferiores.

Paralelamente à criação e aplicação de Mapa de Conceitos no âmbito das Ciências da Educação, surgiu muito mais recentemente no seio das Ciências da Informação um outro instrumento de representação do conhecimento designado por *Topic Maps*.

A norma *ISO 13250 Topic Maps* [12] é um formalismo para a representação de conhecimento sobre a estrutura de um determinado sistema de informação. Qualquer conceito por

mais abstrato ou concreto que seja representa um tópico. Os tópicos podem ter associações com outros tópicos e podem possuir ocorrências (instâncias concretas do lado do sistema de informação). A informação associada a um tópico é calculada em função das associações e ocorrências desse tópico. O conjunto de tópicos, associações e ocorrências é um *topic map*.

Os *Topic Maps* são suficientemente abstratos para especificar qualquer coisa e suficientemente formais para que seja possível a criação de ferramentas de processamento e de navegação. Desta forma, o sistema que se apresenta na seção seguinte utiliza a norma *ISO 13250 Topic Maps* para a gestão do conhecimento encontrado nos documentos didáticos editados por um docente.

III. O SISTEMA *E-NSINO*

Hoje em dia, os docentes se utilizam das mais variadas ferramentas para a produção de seus documentos didáticos, como editores de textos e apresentações convencionais. Contudo, essas ferramentas não foram projetadas especificamente para a produção de documentos didáticos; neles podem ser produzidos quaisquer tipo de documentos. Devido a este fato, os documentos gerados por tais ferramentas acabam por possuir formas distintas entre si, justamente por não haver uma preocupação com a estruturação da informação contida em tais documentos.

A necessidade de haver uma estruturação da informação nos documentos didáticos se dá no momento em que o docente necessita fazer pesquisas sobre os seus documentos. Por exemplo, supondo que o docente queira saber quais são as questões que abordem o tema *História* em suas dezenas de provas aplicadas ao longo de sua carreira acadêmica, uma simples procura através de um motor de busca (como o encontrado nos sistemas operacionais) não será o suficiente, pois tais motores de busca retornam apenas se as palavras foram mencionadas no texto, sem preocupar-se com o contexto das mesmas.

Além disso, possuindo uma estruturação padrão para os principais tipos de documentos didáticos, pode-se separar o conteúdo dos documentos de sua formatação, sendo possível desenvolver processadores responsáveis por produzir diversos formatos de saída para o mesmo tipo de documento. Desta forma, o docente preocupa-se somente com o conteúdo do material didático que está desenvolvendo; posteriormente, no momento de disponibilizar tal material, ele define qual será o formato de saída (HTML, RTF, PDF, LaTeX, TXT, ...) e sua própria formatação (fontes, tamanhos, parágrafos, ...), pois o sistema todo está baseado em XML (*Extensible Markup Language*) [16].

Visando resolver estas questões, o presente projeto destina-se à construção de um sistema para apoio à produção de conteúdos para docentes. Assim, os materiais didáticos foram divididos em três grupos: provas, apostilas e apresentações (*slides*). Desta forma, cada um destes tipos de

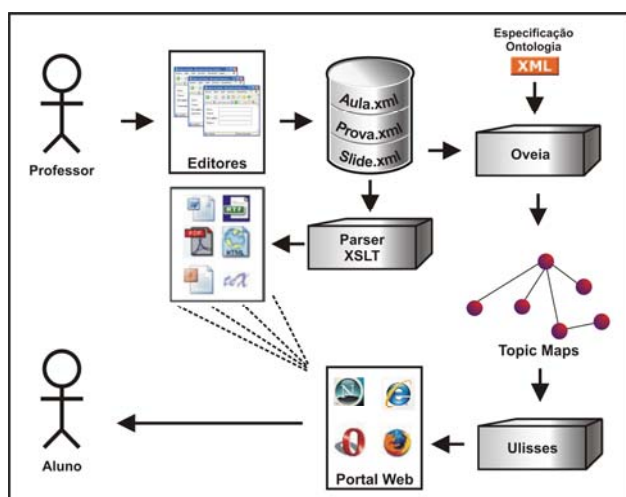


Fig. 1. Arquitetura funcional do sistema *e-Nsino*.

documentos foi estruturado de acordo com as necessidades definidas por um grupo de docentes participantes do projeto.

Para a organização do conhecimento encontrado nos documentos didáticos, optou-se pela utilização de ontologias [13]. A partir da camada semântica representada por uma ontologia, torna-se possível a geração de páginas Web para cada uma das disciplinas, contendo os documentos relacionados entre si através dos seus temas. Com a uma interface Web gerada automaticamente, os alunos tem acesso a todo o conteúdo da disciplina, organizado através da própria ontologia.

A. Arquitetura do sistema *e-Nsino*

A arquitetura funcional do sistema proposto pode ser dividida em quatro grandes componentes, os quais são descritos a seguir e são mostrados na Figura 1.

A Figura 1 apresenta a idéia aqui defendida. Os docentes editarão seus materiais didáticos em editores personalizados, especificamente criados para cada tipo de material didático. Os documentos editados são armazenados em formato XML, próprio para o intercâmbio entre aplicações e processamento de documentos, de acordo com um dos seguintes esquemas:

- *Xlessons*: para a produção de documentos gerais, tais como: guia de aula teórica, guia de aula laboratorial, folhas de exercícios, apostilas, artigos, ...;
- *Xexams*: para a produção dos documentos dos exames referentes aos exames e provas;
- *Xslides*: para a produção dos documentos referentes às apresentações (*slides*).

A partir do conjunto de documentos XML gerados pelos editores, o sistema permite a geração de vários tipos distintos de documentos, tais como PDF, RTF, HTML, TXT e LaTeX, fazendo uso de processadores XSL (*eXtensible Stylesheet Language*) especializados. O armazenamento físico dos documentos produzidos é em uma sub-árvore no sistema de diretórios do servidor.

Os documentos XML gerados pelos editores também servirão como fonte de entrada no *Oveia* [4]. A partir destas fontes XML, o *Oveia* então encarrega-se de extrair um *topic*

map no formato XTM (*XML Topic Maps*) [9], o qual irá representar todos os conceitos contidos nestes documentos, assim como os relacionamentos entre estes conceitos.

Com o *topic map* gerado pelo *Oveia*, a ferramenta *Ulisses* [4] produz um conjunto de páginas Web geradas automaticamente a partir da ontologia expressa no *topic map* que vai permitir o acesso aos documentos produzidos pelo docente.

IV. PRODUÇÃO DE DOCUMENTOS DIDÁTICOS

Todas as aplicações para a produção de documentos partilham a mesma filosofia: baseiam-se nos princípios da documentação estruturada e utilizam anotação descritiva para estruturar os documentos que são produzidos. Esta metodologia tem grandes implicações. A primeira é a de que o formato não é livre, tem uma representação textual, a sua estrutura é completamente descrita através de anotações e está formalmente especificada em uma gramática formal.

As vantagens na utilização da anotação descritiva estão bem documentadas na bibliografia [6] (cap. 2, pág. 14) mas pode-se realçar as mais importantes: portabilidade, reuso, reconfiguração do aspecto visual e longevidade, pois a informação está em um formato neutro e de fácil processamento.

Um documento XML tem uma estrutura lógica à qual corresponde uma hierarquia de elementos. Cada elemento é diferenciado dos restantes através de anotações que são adicionadas ao documento. Nesta perspectiva, um documento é composto por dois tipos de informação: dados e anotações.

Tendo em mãos um documento XML, pode-se transformá-lo em qualquer outro tipo de documento através de transformações com XSL (*Extensible Stylesheet Language*). A idéia fundamental do uso casado de XML e XSL é separar conteúdo de forma de apresentação. O conteúdo em XML possui marcadores muito simples, que podem ser lidos, tanto por humanos como por máquinas. XSL permite especificar como se quer que o texto em XML seja mostrado ao usuário (cores e tamanhos de fontes, tabulações, etc.) assim como o formato dos arquivos.

Sendo assim, adotou-se o formato XML para a representação dos documentos didáticos deste sistema de produção de conteúdos e XSL para as transformações destes documentos para os formatos desejados pelo docente.

A. Tipos de Documentos Didáticos

No âmbito de uma sala de aula, os principais documentos didáticos são: provas, apresentações e apostilas. As próximas subseções tratam, de forma individual, cada um destes tipos de documentos.

Provas: No contexto do sistema *e-Nsino*, as provas geradas pelo docente são consideradas provas estáticas, ou seja, elas são apenas mostradas como um documento gerado para a impressão em papel ou para sua disponibilização na Web.

As provas podem conter diversas questões, as quais podem ser divididas em:

- *Múltipla escolha*: uma questão tem um conjunto de respostas; o aluno tem que indicar a que está correta;
- *Verdadeiro e Falso*: cada questão tem duas possíveis respostas; o aluno tem que indicar, para cada questão, se a resposta é verdadeira ou falsa;
- *Desenvolvimento*: para cada questão, o aluno tem à sua disposição uma área em branco, onde deverá desenvolver a sua resposta à questão.
- *Completar*: no enunciado de cada questão, existem algumas lacunas que devem ser preenchidas pelo aluno.
- *Relacionar*: em cada questão, duas colunas deverão ser associadas.

As questões de cada prova não precisam pertencer rigorosamente a apenas um destes tipos. Para se conseguir isto, cada questão é tratada individualmente, tornando-se na unidade base para a composição das provas.

Abaixo apresenta-se a Gramática Livre de Contexto (GLC) da linguagem *Xexams*:

```
Exam ::= head body footnote?
Head ::= student date student_grade? course? area?
        Discipline code_discipline? Teacher
exam_rate image? observation? ano? institution
Body ::= question comment?
Question ::= (true_false | written | multiple_choice |
        associate | complete)+
True_false ::= direction affirmation image?
Direction ::= (item | noitem)+
Item ::= subitem?
Subitem ::= subsubitem?
Affirmation ::= (text answer justification?)+
Text ::= (item | noitem)+
Written ::= direction answer+ image?
Multiple_choice ::= direction affirmation_choice?
        alternatives+ answer image?
Associate ::= direction+ affirmation_associate image?
Affirmation_associate ::= (column1, column2)+
Complete ::= direction+ affirmation complete+
Affirmation_complete ::= answer*
Footnote ::= message institution?
Image ::= uri, format, label
```

A partir desta GLC, foi construído o XML Schema correspondente, o qual fará a validação sintática dos documentos XML que criados pelos editores referente às provas. Para dar a ideia de uma prova definida na aplicação *Xexams*, mostra-se abaixo um pequeno excerto:

```
<questions>
<question numberQ="Q1">
<description>
<para>A cidade de Portugal onde foi o XATA em 2007 é:</para>
</description>
<choices>
<choice answer="false">
<para>Porto;</para>
</choice>
<choice answer="false">
<para>Braga;</para>
</choice>
<choice answer="true">
<para>Lisboa;</para>
</choice>
</choices>
</question>
```

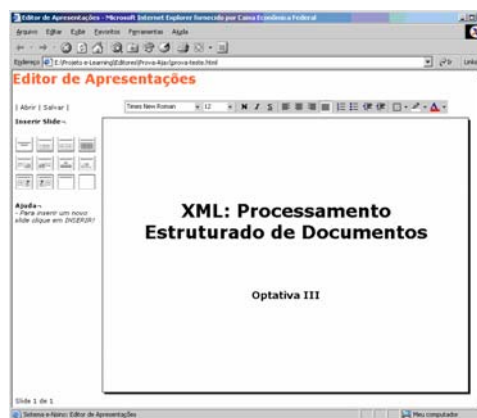


Fig. 2. Interface do editor de apresentações.

```
...
</questions>
```

Neste exemplo, define-se apenas uma questão do tipo booleano. Entretanto, após a tag `</question>`, outras questões podem ser incluídas neste documento, no lugar do "...".

Ou seja, os elementos `<questions>` são compostos por um ou mais elementos `<question>`, os quais contém cada uma das questões da prova.

Apresentações (slides): A aplicação *Xslides* foi desenvolvida para estruturar os documentos referentes às apresentações (*slides*), seguindo a mesma filosofia descrita na seção anterior.

Cada lâmina de uma apresentação pode ser estruturada de diversos formatos, tais como: texto, itens, parágrafos simples, figuras, imagens, misturas de imagens com texto, texto em colunas, entre outros. Com isso, garante-se que os principais formatos de lâminas podem ser construídos pelo docente.

Assim como na subseção anterior, inicialmente definiu-se uma GLC referente ao *Xslides* para, em seguida, gerar um XML Schema correspondente a este tipo de documento, onde expressa-se as regras formais e as restrições quanto aos tipos de dados e os valores aceitos em cada campo.

Aulas práticas e aulas laboratoriais: Este é, talvez, o tipo de documento mais complexo no ambiente aqui proposto. Isso porque uma apostila pode ter inúmeros tipos de construções como, por exemplo: parágrafos; figuras; tabelas; lista de itens numeradas e não-numeradas; palavras em negrito, itálico e sublinhado; *links*; marcações XML; blocos de código; entre outros.

Para atingir esse objetivo, o *DocBook* [14] foi adotado como base, pois o mesmo foi projetado tendo em mente a escrita de documentos estruturados usando SGML ou XML, o qual é apropriado para escrever livros técnicos e artigos. O *DocBook XML* é definido como um conjunto de definições e regras que descrevem exatamente como um documento *DocBook* válido deve ser estruturado. Este formato está se tornando um padrão de fato para documentos técnicos e é suportado por um crescente número de aplicações e ferramentas.

Assim sendo, criou-se uma versão resumida do padrão *DocBook* com seus principais construtores, acrescentando e alterando-se algumas particularidades deste padrão de acordo com as necessidades deste projeto.

Assim como os outros dois tipos de documentos didáticos apresentados anteriormente, também definiu-se previamente uma GLC para a linguagem *XLessons* e posteriormente foi definido um XML Schema para a geração de aulas práticas e aulas laboratoriais. A partir desse XML Schema tornou-se possível a construção dos editores personalizados, dos processadores para geração de formatos de saída e também dos extratores de ontologia destes documentos que seguem o esquema aqui definido. No momento, está-se utilizando essa ferramenta em aulas, suportando assim as aulas práticas e sessões laboratoriais, onde as apostilas fornecidas em papel são substituídas por conjuntos de páginas Web acessíveis pela Internet (o que poderá causar outro tipo de questões relacionadas com os direitos autorais, fato este que está sendo analisado).

B. Edição dos Documentos Didáticos

A geração dos documentos didáticos é efetuada a partir de editores especializados criados especificamente para o esquema XML definido para cada um dos três tipos de documentos apresentados anteriormente. Desta forma, o resultado da edição realizada em cada editor será um documento XML que conterá toda a informação referente à prova, à apostila ou à apresentação criada, de acordo com o editor utilizado.

Com objetivo de proporcionar ao usuário do sistema uma interface onde o mesmo possa desenvolver seu material de aula – provas, apostilas e apresentações – foram criados editores personalizados para cada tipo de material específico.

Para o desenvolvimento dos editores foi utilizada a linguagem PHP, o Sistema Gerenciador de Banco de Dados MySQL juntamente com Ajax, gerando como saída documentos XML a fim de tornar os editores flexíveis quanto a questão de portabilidade, e com uma interface amigável e funcional para o usuário.

Após a edição do material didático, será gerado um documento XML que respeita a estrutura formalizada pelo XML Schema referente ao material editado em questão.

Na edição de uma prova, o professor tem a disposição uma lista com os cinco possíveis tipos de questões onde ele fará a escolha da questão que deseja inserir na prova, as questões podem ser inseridas de forma aleatória, conforme a necessidade do professor. Uma vez escolhida a questão e inserida na área de edição da prova, o professor digitará as informações referentes a questão conforme o tipo de questão escolhida. Também haverá a possibilidade de se excluir uma questão e alterar o tipo da mesma. Ainda para auxiliar na formatação da prova, o professor também contará com uma barra de formatação, onde o mesmo poderá alterar o tipo de fonte, cor, tamanho e disposição do texto, além de colocar o texto em negrito, itálico e/ou sublinhado.

No editor de apostilas, o docente poderá criar as suas apostilas conforme suas necessidades, tendo a sua disposição ferramentas que o permitirão tornar o conteúdo mais atrativo. Ele poderá inserir imagens, além de poder formatar o texto da forma que lhe pareça mais conveniente. Para tanto, ele contará com uma barra de formatação, onde poderá alterar o tipo de fonte, cor, tamanho e disposição do texto, além de colocar o texto dos slides em negrito, itálico e/ou sublinhado, entre outras formatações.

Tal como nos outros editores, na edição de apresentações o docente também contará com uma barra de formatação, onde o mesmo poderá alterar o tipo de fonte, cor, tamanho e disposição do texto, além de colocar o texto dos slides em negrito, itálico e/ou sublinhado, para citar apenas as principais. A Figura 2 dá uma noção da interface do editor de *slides*, a qual é similar aos dois editores introduzidos anteriormente.

A apresentação na Figura 2 é transformada pelo editor em um documento XML que segue o esquema de *Xslides*, conforme apresentado no pseudocódigo abaixo:

```
<?xml version="1.0" encoding="UTF-8"?>
<slides xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Xslides.xsd">
  <slide>
    <layout_texts>
      <title_subtitle>
        <title>XML: Processamento Estruturado de
          Documentos</title>
        <subtitle>Optativa III</subtitle>
      </title_subtitle>
    </layout_texts>
  </slide>
</slides>
```

C. Conversão de Formatos

Após a edição de um documento, o mesmo é salvo pelo editor em um formato XML previamente descrito. Desta forma, utiliza-se a linguagem XSL para seu processamento. De acordo com as vantagens de se utilizar XSL descritas no princípio desta seção, optou-se por construir um lote de folhas de estilos XSLT e XSL:FO para a conversão de cada documento XML gerado pelos editores para os seguintes formatos:

- **HTML:** Para cada documento didático, é gerado uma página Web. No caso das apresentações, cada lâmina corresponde a uma página; a apresentação conterá ainda um índice e a capa;
- **PDF:** Para a produção dos documentos PDF, é utilizado o processador Apache FOP e folhas de estilo XSL:FO;
- **LaTeX:** No âmbito do sistema *e-Nsino*, é utilizado o *Beamer* (uma classe LaTeX) para composição de apresentações, as quais terão o formato PDF ou PS;
- **RTF:** Com documentos no formato RTF, o docente pode utilizar seus documentos em editores de texto que trabalham com este formato.
- **TXT:** Geração de documentos textuais puros.

A escolha destes cinco formatos acima citados foi baseada nos formatos abertos mais utilizados para a produção e

armazenamento de documentos didáticos. Contudo, para acrescentar um novo formato de saída para o ambiente *e-Nsino* basta desenvolver um novo conjunto de folhas de estilos XSL para transformar os documentos (que obedecem os esquemas XML dos documentos didáticos) em outro formato qualquer.

Os processadores e as correspondentes folhas de estilos XSL são acionados automaticamente pelo docente, de forma que após a conclusão da edição de seus documentos, os mesmos são salvos no formato XML e transformados para os formatos acima citados, de forma transparente para o docente.

V. GERAÇÃO DAS PÁGINAS WEB

Com o conjunto de documentos XML gerados pelo docente ao operar os editores especializados, o próximo passo é a geração automática das páginas Web a serem disponibilizadas aos alunos. Contudo, estas páginas criadas correspondem a uma organização abstrata e virtual dos documentos produzidos, cuja organização abstrata é dada através da especificação de uma ontologia.

Dos vários formalismos e modelos inerentes possíveis para a representação de ontologias para a Web, foi selecionada a norma *ISO 13250 Topic Maps* [12] devido ao seu elevado grau de abstração e independência aos dados, o que os torna ideais para quem quer gerar ferramentas automáticas.

Assim, para a criação, armazenamento e processamento da ontologia que representa o conhecimento extraído sobre o material didático, são utilizados dois módulos do ambiente *Metamorphosis* [4]: o *Oveia* e o *Ulisses*. O *Metamorphosis* é um ambiente para extração, validação, armazenamento e navegação de *Topic Maps*, composto por três módulos: *Oveia*, *XTche* e *Ulisses*.

Resumidamente, a idéia principal é extrair um pequeno conjunto de metadados de cada fonte de informação, construir uma rede semântica de conceitos e relações entre conceitos e povoá-la com os metadados previamente extraídos das fontes. Mais tarde, esta base de conhecimento é utilizada para guiar semanticamente o usuário na navegação até chegar aos dados que efetivamente lhe interessam.

A. Criação da Ontologia com o Oveia

O *Oveia* é um extrator de ontologias representadas em *Topic Maps*. Sua arquitetura é composta por duas especificações e os referentes processadores [4]: a primeira, escrita na linguagem XSDS (*XML Specification for DataSources/DataSets*), especifica os dados a serem extraídos das fontes de informação; enquanto que a segunda, escrita na linguagem XS4TM (*XML Specification for Topic Maps*), é responsável por definir a ontologia a ser gerada. Com base nestas especificações, o extrator busca as informações nas fontes de informação e produz um *topic map* representado no formato XTM.

No caso do *e-Nsino*, ambas especificações (XSDS e XS4TM) são produzidas e atualizadas no momento da edição de cada documento pelo docente. Desta forma, quando o docente deseja criar as páginas Web referente às suas aulas,

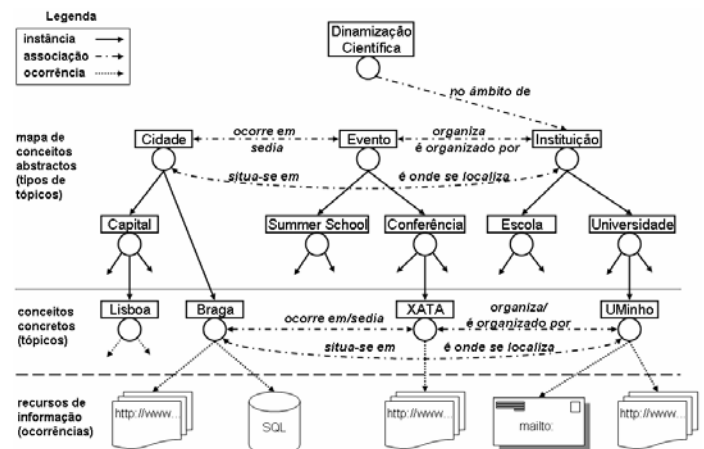


Fig. 3. Ontologia extraída dos documentos didáticos.

basta executar a aplicação em questão, pois tanto os documentos XML e as especificações estarão disponíveis. O resultado deste processamento é um documento XTM contendo a ontologia extraída das fontes de informação editadas pelo docente.

Tomando como base os documentos *Xslides*, *Xlessons* e *Xexams* apresentados na seção IV, o sistema *e-Nsino* efetua a extração automática da ontologia representada nos documentos editados pelo docente e a representa de acordo com o formato XTM. A Figura 3 apresenta uma visão dessa representação.

Como se pode perceber na Figura 3, o *topic map* serve como um índice, o qual indexa todos os documentos encontrados no sistema de acordo com os temas aos quais cada documento versa. Assim, todos os temas são representados por páginas, as quais estão relacionadas entre si através de associações. Por fim, as páginas possuem ocorrências que apontam para os documentos correspondentes.

B. Geração de Interfaces Web com o Ulisses

O sistema *e-Nsino* faz uso da ferramenta *Ulisses* para a criação das páginas Web. O *Ulisses* é uma ferramenta para visualização de *topic maps* baseada em conceitos de navegação em grafos, permitindo uma navegação na sua rede conceitual, além de acessar todos os recursos de informação apontados pelas ocorrências. O resultado é a criação de páginas Web a partir de um *topic map*, como o gerado pelo *Oveia*.

A idéia sobre a qual se baseou o *Ulisses* é a idéia da navegação conceitual, a qual pode ser descrita como: quando se está posicionado sobre um certo tema, a ferramenta de navegação mostrará as informações associadas a este tema em particular; se for escolhido algum dos outros temas relacionados, a navegação muda para a visão deste novo tema; se for escolhido algum dos recursos de informação, o sistema mostrará o conteúdo do próprio recurso.

O *Ulisses* fornece uma navegação completa sobre *topic maps*. A representação do conhecimento é apresentado de uma forma simples e precisa, formando uma rede semântica baseada em tópicos e associações. Quando se acessa às



Fig. 4. Navegação no *Ulisses*.

informações referente a um determinado tópico no *Ulisses* como na Figura 4, visualizam-se suas características (o seu tipo, suas instâncias, seus identificadores de tema, seus nomes e suas ocorrências) e as associações relacionadas com este tópico (incluindo os papéis de associação atuado por ele e os tópicos associados) [11].

A visão apresentada na Figura 4 é uma página Web que está situada no nodo correspondente ao tema *XML Topic Map Builder*, um artigo apresentado no XATA 2003. Este mesmo *topic map* pode ser visualizado conforme um grafo, como mostra a Figura 5.

Deste modo, o *topic map* é visto como um grafo onde cada tópico (tema) é representado como um nodo do grafo e as associações são representadas pelas relações entre os nodos. Desta forma, pode-se navegar por todo o *topic map*, passando de tópico a tópico, a partir das associações entre os mesmos.

Ao permitir criar um mapa virtual sobre as fontes de informação, esta tecnologia permite-nos obter diferentes vistas refletindo diferentes organizações da informação sem que as fontes de informação sejam alteradas.

VI. AVALIAÇÃO DO AMBIENTE *E-NSINO*

O desenvolvimento de aplicativos para Internet está cada vez maior. Como a quantidade de páginas Web tem aumentado gradativamente, os autores devem ter a preocupação com a eficiência e a qualidade de seus sites, principalmente nos aspectos de usabilidade, acessibilidade e testes.

A. Usabilidade e Acessibilidade

A qualidade de uma boa página Web está relacionada à aparência visual e à estrutura da informação, permitindo que os usuários encontrem o que estão procurando de forma rápida e eficiente. Como a ferramenta proposta gera automaticamente páginas Web com interface gráfica, contendo textos, tabelas, imagens e ilustrações preocupou-se em apresentar páginas atrativas e funcionais considerando os critérios de usabilidade e acessibilidade para Web.

A avaliação de interfaces não deve ser considerada unicamente na etapa de *design*, mas deve ser executada em todo processo de desenvolvimento de um software, fazendo análise dos resultados e melhorias gradativas na interface [5] (cap. 4, pág. 163). Seguindo esta consideração, cabe ressaltar que no processo de desenvolvimento do sistema, a avaliação foi aplicada em paralelo com o desenvolvimento do software.

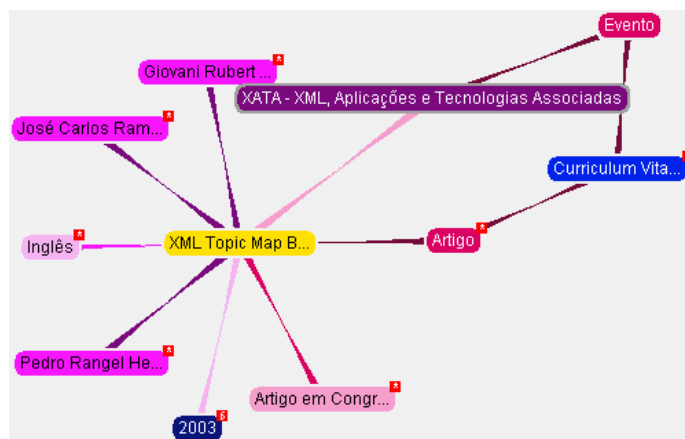


Fig. 5. Mapa Conceitual do *topic map* gerado.

De acordo com o documento da W3C-WAI [17], considerado uma referência para os princípios de acessibilidade e *design*, têm-se como recomendação dois temas genéricos: assegurar uma transformação harmoniosa e tornar o conteúdo compreensível e navegável. Para apresentar uma transformação harmoniosa e tornar o conteúdo compreensível e navegável, o sistema apresenta as seguintes características:

- Separa a estrutura da apresentação;
- Fornece descrição para elementos não textuais;
- Identifica os cabeçalhos de linha e de coluna em tabelas;
- Cria documentos independentes de plataforma;
- Inclui ferramentas de navegação e orientação;

A verificação da usabilidade em sistemas pode ser realizada através de diversos métodos de avaliação. Os objetivos principais desta verificação são: avaliar a funcionalidade, avaliar o impacto do *design* junto ao usuário e identificar problemas específicos com o *design*.

Uma das técnicas de avaliação da usabilidade é baseada em heurísticas. As heurísticas de usabilidade aplicadas neste projeto foram criadas por Nielsen [5] (cap. 4, pág. 170). Segundo o autor, esta técnica é fácil, rápida e barata. Cada avaliador percorre a interface diversas vezes inspecionando os diferentes componentes e ao detectar problemas relata quais heurísticas foram violadas.

Neste projeto aplicou-se este tipo de avaliação, onde em um primeiro momento detectou-se que as seguintes heurísticas foram violadas:

1. Controle do usuário e liberdade: não existiam opções de saída para os casos onde o usuário havia escolhido por engano alguma funcionalidade do sistema, ou seja, falta de funções como *undo* e *redo*;
2. Prevenção de erros: o sistema não previnha possíveis erros cometidos pelo usuário;
3. Flexibilidade e eficiência de uso: o sistema apresentava carências em formas de acesso mais rápida em algumas de suas funcionalidades, não oferecendo aos usuários mais experientes formas de acelerar a interação.
4. *Help* e documentação: a inexistência de um *help* que focalize as principais funcionalidades do sistema e direcione as tarefas do usuário também estava faltando no sistema.

Como já foi descrito anteriormente, a avaliação foi aplicada durante o processo de desenvolvimento do sistema estes problemas identificados através da avaliação com estas heurísticas já foram corrigidos, possibilitando melhorias na interface, para a obtenção da usabilidade.

Cabe ressaltar que não foram diagnosticados problemas na interface do software quando analisado sob as seguintes heurísticas: visibilidade do status do sistema, compatibilidade com o mundo real, consistência e padrões, ajudar os usuários a reconhecer, diagnosticar e corrigir erros, estética e *design* minimalista e reconhecimento ao invés de relembração.

B. Implementação e Testes

A implementação do ambiente *e-Nsino* fez uso de diversas tecnologias, sempre visando a utilização do ambiente em diversas plataformas. Conforme citado anteriormente, todos os editores apresentados na subseção IV-B foram implementados nas linguagens PHP e Ajax, utilizando o banco de dados MySQL para a gestão das contas de usuários e XML para o armazenamento dos documentos. Por sua vez, os *parsers* com a função de converter os documentos didáticos em diferentes formatos foram programados com a linguagem XSL. A implementação dos ambientes *Oveia* e *Ulisses* foi efetuada na linguagem Java.

A atividade de testes é uma etapa indispensável no desenvolvimento de um software. Dada a diversidade dos testes de software existentes, é vantajoso considerar os tipos de testes, à medida que se tornam disponíveis a um projetista. Sendo assim, os principais tipos de testes são [15] (cap. 17):

1. *Testes de Desempenho*: determinam o desempenho do sistema de software como tempo de execução, ou seja, durante o tempo de execução do software associado a várias partes do código, ao tempo de resposta e à utilização do dispositivo.
2. *Testes Funcionais*: são os testes que abordam e enfatizam as entradas, saídas e princípios funcionais dos módulos do software.
3. *Testes Estruturais*: destinam-se ao exercício da lógica interna de um sistema de software.
4. *Testes de Validação*: série final dos testes de software. A validação é bem sucedida, quando o software funciona de uma maneira razoavelmente esperada pelo usuário.
5. *Testes de Aceitação*: são testes formais conduzidos para determinar quando o sistema de software satisfaz os critérios de aceitação e, portanto, é aceito de acordo com as necessidades do usuário.

No *e-Nsino* foram aplicados todos os testes acima descritos, porém a ênfase maior foi dada na aplicação dos Testes de Aceitação, porque trata diretamente com o envolvimento do usuário no desenvolvimento dos requisitos e critérios de aceitação. Portanto, o objetivo deste tipo de teste foi de verificar se o software atendeu às necessidades do usuário.

Os testes foram aplicados com questionários aos usuários com o objetivo de verificar a eficiência e a eficácia do sistema. Os questionários foram respondidos pelos usuários

em seu ambiente de trabalho, sem a presença do desenvolvedor. Após o preenchimento, o software foi corrigido e novamente testado para verificar se os erros foram solucionados.

VII. CONCLUSÃO

O sistema *e-Nsino* surgiu, inicialmente, como uma aplicação prática do ambiente *Metamorphosis*, resultado de uma tese de doutorado [3]. Desta forma, uni-se ontologias com ensino a distância, criando um ambiente que permite ao docente uma edição especializada dos seus documentos didáticos, a geração de vários formatos distintos para cada documento, além de gerar interfaces Web para a disponibilização dos mesmos guiadas através de uma rede semântica.

Uma das principais vantagens do *e-Nsino* é a utilização de XML em todos os seus módulos. Com isso, garante-se a portabilidade e o reuso dos documentos criados, a independência entre *software* e *hardware*, além de facilitar a manutenção e o desenvolvimento de demais tecnologias para serem acrescentadas a este ambiente.

Atualmente, o *e-Nsino* está sendo utilizado por professores dos cursos de Ciência da Computação e Sistemas de Informação do Centro Universitário Franciscano (UNIFRA). Na maioria dos casos práticos, percebe-se que o ambiente tem sido utilizado para disciplinas ministradas a distância, onde o portal Web da mesma é gerado automaticamente pelo próprio ambiente, disponíveis na intranet da instituição. Contudo, em alguns casos o *e-Nsino* serviu como base para a preparação propriamente dita do material didático. Os resultados da utilização do ambiente na prática por docentes serviram para detectar melhorias que já estão empregadas na versão atual do sistema; outras, entretanto, somente serão implementadas na sua próxima versão.

Em termos de trabalhos futuros, um dos projetos é a criação de um sistema para a avaliação automática das provas geradas pelo sistema. Assim, as provas poderão ser dinâmicas a ponto de serem feitas via Web pelos alunos, podendo possuir alguns termos funcionais, tais como questões com tentativa única e limite temporal para a resposta.

Outro objetivo a ser posto em prática na próxima versão do *e-Nsino* é a utilização de documentos didáticos existentes, ou seja, que o docente já tenha criado previamente, fazendo com que ele não tenha a necessidade de editá-los através do próprio ambiente. Para isso, será utilizado o extrator de ontologias *Oveia* [4], o qual permite a extração automática de uma ontologia a partir de um sistema de informação que pode ser composto de bancos de dados e documentos de diversos formatos distintos. Desta forma, qualquer documento didático poderá ser posto na rede semântica de uma disciplina ou curso, independente do formato ou do editor utilizado para sua construção.

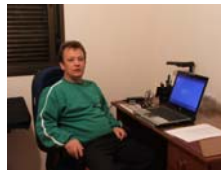
REFERÊNCIAS

- [1] Advanced Distributed Learning, Sharable Content Object Reference Model – SCORM 2004, 2nd Edition – Overview, 2004.
- [2] D. P. Ausubel, *Educational Psychology, A Cognitive View*. New York: Holt, Rinehart and Winston, Inc, 1968.
- [3] G. R. Librelotto, *XML Topic Maps: da Sintaxe a Semantica*, PhD thesis, Departamento de Informática, Escola de Engenharia, Universidade do Minho, 2005.
- [4] G. R. Librelotto and J. C. Ramalho and P. R. Henriques, Metamorphosis - A Topic Maps Based Environment to Handle Heterogeneous Information Resources, In *Lecture Notes in Computer Science*, volume 3873, pages 14–25, Springer-Verlag GmbH., 2006.
- [5] H. V. Rocha and M. C. C. Baranauskas, *Design e Avaliação de Interfaces Humano-Computador*, Campinas, SP: NIED/UNICAMP, 2003.
- [6] J. C. Ramalho and P. R. Henriques, *XML & XSL: Da Teoria a Prática*, FCA Editora, 2002.
- [7] J. Novak and D. Gowin, *Aprendendo a aprender*. Barcelona: Martínez Roca – livros universitários y profesionales, 1988.
- [8] J. Novak and A. Cacas, *The Theory Underlying Concept Maps and How to Construct Them*, Technical Report IHMC CmapTools 2006-01, Institute for Human and Machine Cognition, 2006.
- [9] J. Park and S. Hunting, *XML Topic Maps: Creating and Using Topic Maps for the Web*, ISBN 0-201-74960-2, Addison Wesley, 2003.
- [10] R. Koper and B. Olivier and T. Anderson (Editors), *IMS Learning Design Information Model. Version 1.0*. IMS Global Learning Consortium, 2003.
- [11] L. M. Garshol and G. Moore, *Topic Maps – Data Model*, <http://www.isotopicmaps.org/sam/sam-model/>, 2005.
- [12] M. Biezunsky and M. Bryan and S. Newcomb, *ISO/IEC 13250 – Topic Maps*, ISO/IEC JTC 1/SC34, 1999.
- [13] N. Guarino, *Formal Ontology and Information Systems*. In Conference on Formal Ontology (FOIS98), 1998.
- [14] N. Walsh and L. Muellner, *Docbook: The Definitive Guide*, O'Reilly & Associates, 1999.
- [15] R.S. Pressman, *Engenharia de Software*. 5 ed. Rio de Janeiro: Mc Graw-Hill, 2002.
- [16] W3C, *Extensible Markup Language (XML)*, <http://www.w3.org/XML/>, 2006.
- [17] W3C-WAI, *Web Accessibility Initiative*, <http://www.w3.org/WAI/>, 2006.
- [18] W. Hodgins, *WG12: Learning Object Metadata*, <http://ltsc.ieee.org/wg12/>, 2003.



Giovanni Rubert Librelotto é Doutor em Informática pela Universidade do Minho, Portugal, desde 2005. Em 2001, obteve o título de Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Atualmente é professor titular do Centro Universitário Franciscano (UNIFRA), em Santa Maria, RS. É professor dos cursos de Mestrado em Nanociências e de Graduação de Ciência da

Computação e Sistemas de Informação. Publicou diversos artigos em periódicos especializados e em anais de eventos nacionais e internacionais. Atualmente coordena 2 projetos de pesquisa. Suas principais áreas de interesses são: eXtensible Markup Language (XML), Topic Maps, Compiladores, Semantic Web, Constraint Languages e Linguagens de Programação.



Marcos Luís Cassal possui graduação em Informática pela Universidade Regional do Noroeste do Estado do Rio Grande do Sul (1995) e mestrado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (2001). Atualmente é professor titular do Centro Universitário Franciscano. Atua principalmente

nos seguintes temas: Computação Gráfica, Geometria Sólida Construtiva - CSG, Realismo, OpenGL.