

e-Nsino: Um Sistema de Auxílio à Produção e Gestão de Documentos Didáticos

**Giovani Rubert Librelotto and Ricardo Nenê and Maurício Friedrich
and Rogério Turchetti and Marcos Luís Cassal**

¹UNIFRA – Centro Universitário Franciscano
Rua dos Andradas, 1614 – 97010-032 – Santa Maria – RS – Brasil

{giovani,turchetti,cassal}@unifra.br

***Resumo.** Este artigo apresenta 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 gerada de forma automática a partir da representação do conhecimento extraída dos documentos editados pelo docente.*

***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 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.*

1. Introdução

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. Tais ferramentas 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.

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 Brasileira* 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*) [Ramalho and Henriques 2002].

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 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 [Guarino 1998]. 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. A partir de uma interface Web gerada automaticamente, os alunos tem acesso a todo o conteúdo da disciplina, organizado através da própria ontologia.

O presente artigo está estruturado da seguinte forma: a Seção 2 apresenta o sistema e-Nsino, com ênfase na estrutura deste sistema. A estruturação, edição e transformações dos documentos didáticos são abordadas na Seção 3. A Seção 4 descreve a extração da ontologia e a produção das páginas Web referentes aos documentos didáticos. Conclui-se o artigo na Seção 5 apresentando seu sumário seguido dos trabalhos futuros.

2. O sistema e-Nsino

O sistema e-Nsino destina-se como uma ferramenta para apoio à produção de documentos didáticos. Efetivamente, ao analisar com cuidado os produtos mais utilizados para a criação de documentos didáticos, constata-se que estes permitem construir praticamente tudo. No entanto, estes sistemas não definem uma tecnologia, ou formato, a ser adotado para a produção dos conteúdos didáticos de forma a possibilitar uma uniformização de tais documentos. Foi para preencher esta lacuna que o e-Nsino foi pensado.

2.1. 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 criarão seus documentos em editores personalizados, especificamente projetados 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 XML:

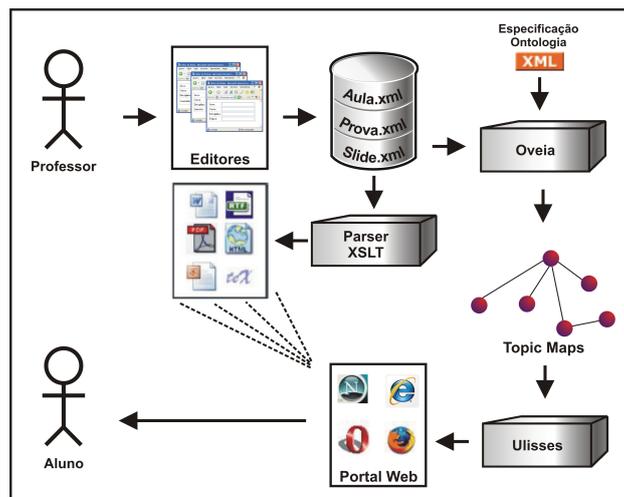


Figure 1. Arquitetura funcional do sistema e-Nsino

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 formatos distintos de documentos, fazendo uso de processadores XSL (*eXtensible Stylesheet Language*) [Harold and Means 2001] especializados.

Os documentos XML gerados pelos editores também servirão como fonte de entrada no Oveia¹ [Librelotto et al. 2006]. A partir destas fontes XML, o Oveia então encarrega-se de extrair um *topic map* no formato XTM (*XML Topic Maps*) [Park and Hunting 2003], 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 [Librelotto et al. 2006] 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. Mais detalhes sobre essas ferramentas serão encontrados na Seção 4.

3. 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.

As vantagens na utilização da anotação descritiva estão bem documentadas na bibliografia [Ramalho and Henriques 2002] [Harold and Means 2001] 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

¹O Oveia é um extrator automático de ontologias representadas no formato ISO 13250 Topic Maps.

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. 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, tamanhos de fontes, etc – assim como o formato dos arquivos.

3.1. 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.

3.1.1. 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 diversos tipos: múltipla escolha, verdadeiro e falso, dissertativas, para completar e também para relacionar colunas. 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, formate, 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.

3.1.2. 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, ítems, parágrafos simples, figuras, imagens, misturas de imagens com texto, texto em colunas, entre outros. A fim de formalizar a linguagem **Xslides**, definiu-se a GLC abaixo:

```
Slides ::= opening body
Opening ::= title subtitle? authors institution? place? date?
Authors ::= author+
Author ::= name email?
Body ::= slide+
Slide ::= (subtitles | texts | columns | contents)
Subtitles ::= subtitle?
Texts ::= text
Text ::= (item | noitem+)
Item ::= subitem*
Subitem ::= subsubitem*
Columns ::= column_text column_content
Column_text ::= (item | noitem+)
Column_content ::= (image | table) label?
Image ::= uri formate
Contents ::= (image | table) label?
```

Assim como na subseção anterior, obteve-se um XML Schema correspondente à linguagem **Xslides**, onde expressa-se as regras formais e as restrições quanto ao tipo de dados e os valores aceitos em cada campo.

3.1.3. 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 ítems numeradas e não-numeradas; palavras em negrito, itálico e sublinhado; *links*; marcações XML; blocos de código; entre outros.

Para atingir esse objetivo, o DocBook [Walsh and Muellner 1999] foi adotado como base, pois o mesmo foi projetado tendo em mente a escrita de documentos estruturados usando XML, sendo apropriado para escrever livros técnicos e artigos. 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-se algumas particularidades deste padrão de acordo com as necessidades deste projeto. A GLC a seguir mostra a estrutura formal de **Xlessons**:

```
Book ::= bookinfo chapter+
Bookinfo ::= title subtitle? date author+ abstract? publishing_company? isbn? edition?
Date ::= day month year
Author ::= firstname surname? lastname email?
Abstract ::= title para
Para ::= (link | email | markup | emphasis | br)?
Publishing_company ::= name address
Chapter ::= title para+ section?
Section ::= title (para | image | table)*
Emphasis ::= item noitem
Table ::= title tgroup
Tgroup ::= thead tbody
Address ::= street_avenue number complement? quarter zip_code? post_office? city, state
country fone email? fax? site?
```

Pretende-se utilizar essa ferramenta em aulas, suportando assim as aulas práticas e sessões laboratoriais, onde as apostilas fornecidas em papel são substituídas por con-

juntos de páginas Web acessíveis pela Internet (o que pode causar outro tipo de questões relacionadas com os direitos autorais, fato este que está sendo analisado).

3.2. 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.

Neste momento, os editores são baseados na ferramenta *Altova Authentic*². O *Authentic* é uma ferramenta de uso gratuito produzido pela *Altova* para a edição de documentos XML. Ele simplifica a edição de documentos XML através de formulários eletrônicos usando uma interface do estilo de um processador de texto.

Desta maneira, criou-se formulários específicos através da ferramenta *Altova StyleVision*³ para cada um dos tipos de documentos no contexto do e-Nsino. Assim, o docente utiliza o *Authentic* onde carregará o estilo referente ao tipo de documento desejado para efetuar as edições de seus documentos.

3.3. 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 FOP⁴ e folhas de estilo XSL:FO;

LaTeX: No âmbito do sistema e-Nsino, é utilizado o *Beamer*⁵ (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 ferramentas do tipo *Office*.

TXT: Geração de documentos textuais puros.

Os processadores e as correspondentes folhas de estilos XSLT e XSL:FO 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.

²http://www.altova.com/products/authentic/xml_db_form_editor.html

³http://www.altova.com/products/stylevision/xslt_stylesheet_designer.html

⁴<http://xmlgraphics.apache.org/fop/>

⁵<http://latex-beamer.sourceforge.net/>

4. 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* [Biezunsky et al. 1999] devido ao seu elevado grau de abstração e independência aos dados, o que os torna ideais para quem quer gerar ferramentas automáticas.

Os *Topic Maps* [Park and Hunting 2003] são um formalismo para a representação de conhecimento sobre a estrutura de um determinado sistema de informação. Eles são suficientemente abstratos para especificar qualquer coisa e suficientemente formais para que seja possível a criação de ferramentas de processamento e de navegação. Assim, para a criação, armazenamento e processamento da ontologia que representa o conhecimento [Sowa 2000] extraído sobre o material didático, são utilizados o *Oveia* e o *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.

4.1. Criação da Ontologia com o Oveia

O *Oveia* é um extrator de ontologias representadas no padrão *ISO 13250 Topic Maps*. Sua arquitetura é composta por duas especificações e os referentes processadores [Librelotto et al. 2006]: a primeira, escrita na linguagem *XSDS (XML Specification for DataSources/DataSets)*, especifica os dados a serem extraídos das fontes de informação; enquanto que 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, basta executar a aplicação em questão, pois tanto os documentos XML e as especificações estarão disponíveis. O resultado deste processamento é um documento no formato *XTM* contendo a ontologia extraída das fontes de informação editadas pelo docente.

4.2. 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 do processamento do *Ulisses* é a criação de um conjunto de páginas Web a partir de um *topic map* representado no formato *XTM*, como o gerado pelo *Oveia*.

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.

5. Conclusão

O sistema e-Nsino surgiu, inicialmente, como uma aplicação prática sobre o resultado de uma tese de doutorado sobre *Topic Maps* [Librelotto 2005]. Desta forma, uniu-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.

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.

References

- Biezunsky, M., Bryan, M., and Newcomb, S. (1999). ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34. <http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>.
- Guarino, N. (1998). Formal Ontology and Information Systems. In *Conference on Formal Ontology (FOIS98)*. <http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/FOIS98.pdf>.
- Harold, E. R. and Means, W. S. (2001). *XML in a Nutshell*. O'Reilly & Associates.
- Librelotto, G. R. (2005). *XML Topic Maps: da Sintaxe à Semântica*. PhD thesis, Departamento de Informática, Escola de Engenharia, Universidade do Minho.
- Librelotto, G. R., Ramalho, J. C., and Henriques, P. R. (2006). 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.
- Park, J. and Hunting, S. (2003). *XML Topic Maps: Creating and Using Topic Maps for the Web*, volume ISBN 0-201-74960-2. Addison Wesley.
- Ramalho, J. C. and Henriques, P. (2002). *XML & XSL Da Teoria à Prática*. FCA Editora.
- Sowa, J. F. (2000). *Knowledge Representation: logical, philosophical and computational foundations*. Brooks/Cole.
- Walsh, N. and Muellner, L. (1999). *Docbook: The Definitive Guide*. O'Reilly & Associates.