

Meta-Objetos SCORM Hierárquicos via Reticulados Conceituais

Luciano Silva, Denise Stringhini e Ismar Frango Silveira
Faculdade de Computação e Informática, Universidade Mackenzie
Rua da Consolação, 930 - 01302-907 São Paulo, Brasil
{lucianosilva,dstring,ismar}@mackenzie.br

Resumo. SCORM é um conjunto de padrões e especificações que disponibilizam facilidades para interoperabilidade, acessibilidade e reuso para ambientes de *e-learning* baseados em Web. Apesar de ser bastante geral, o modelo SCORM ainda carece de mecanismos de acoplamento entre objetos para formação de redes conceituais. Estas redes normalmente são limitadas às seqüências lineares impostas pelas Árvore de Atividades e acessadas pelos Sistemas Gerenciadores de Aprendizado para controlar a navegação. Este trabalho apresenta uma arquitetura alternativa para acoplamento entre objetos SCORM, baseada na teoria dos reticulados conceituais e tecnologia XLink, que permite a construção de meta-objetos SCORM com estrutura conceitual hierárquica e não-linear.

1 Introdução

Objetos de aprendizagem [Wiley 2000] representam uma área de intensa pesquisa em ambientes de ensino-aprendizagem mediado por computador, principalmente naqueles baseados em Web. Existem diversas propostas de padrões para interoperabilidade, acessibilidade e reuso destes objetos entre as várias alternativas de Sistemas Gerenciadores de Aprendizagem (LMS – *Learning Management Systems*).

Um padrão bastante difundido para atender aos requisitos levantados é denominado SCORM (*Sharable Content Object Reference Model*). Há vários outros padrões de metadados disponíveis para Objetos de Aprendizagem, como o Dublin Core [DCMI 2004], IMS [IMS 2004] e LOM [IEEE 2005]. Contudo, SCORM tende a tornar-se o padrão globalmente aceito, dado o atual suporte recebido por repositórios de objetos de aprendizagem. Tal padrão é construído sob três documentos básicos: Modelo de Agregação de Conteúdo, Ambientes de Execução, Sequenciamento e Navegação. O Modelo de Agregação de Conteúdo permite realizar anotações quantitativas e qualitativas sobre o objeto de aprendizagem, o Ambiente de Execução define o ambiente operacional necessário para a execução do objeto e o Sequenciamento e Navegação define uma ordem linear para exibição de vários objetos de aprendizagem.

O documento de Sequenciamento e Navegação, em particular, é bastante discutido na literatura, principalmente quanto às limitações das Árvore de Atividades que, em linhas gerais, definem a navegação nos objetos. O esquema de navegação definido pelas árvores é interpretado pelo LMS como a ordem de percurso e, limitações apresentadas pelo esquema, também limitam a navegação do aprendiz no LMS.

Este trabalho apresenta a proposta de uma arquitetura de navegação entre objetos SCORM baseado na Teoria dos Reticulados Conceituais, uma maneira bastante natural para especificar objetos de aprendizagem. São propostas introduções de anotações e *links* com alta estrutura semântica, via tecnologia XLink, nos objetos SCORM, o que permite tratamento personalizado de navegação ponto-a-ponto entre os objetos e navegação não-linear. A organização do trabalho é descrita a seguir: a Seção 2 apresenta a Teoria das Hierarquias e Reticulados Conceituais; a Seção 3 retoma os principais conceitos do padrão SCORM e suas principais limitações de navegação; as Seções 4 e 5 apresentam a propostas de anotações nos objetos e nos links e, finalmente, a Seção 6 apresenta as conclusões e trabalhos futuros.

2 Hierarquias e Reticulados Conceituais

A teoria de conjuntos parcialmente ordenados e reticulados [Davey e Priestley 2002] aplicada ao estudo de hierarquias tem produzido várias contribuições nas mais diversas de computação como Inteligência Artificial, Teoria das Categorias, Semântica de Linguagens de Programação e Teoria de Concorrência.

Esta seção fará uma breve incursão na área denominada Análise de Conceitos Formais, onde grande parte do potencial de Teoria de Reticulados tem sido utilizada para fins de aprendizado hierárquico de conceitos.

2.1 Conceitos e Contextos

A definição de conceito é uma questão filosófica bastante complexa. De uma maneira bastante geral, um *conceito* é determinado pela sua *extensão* e *intensão*. Uma extensão consiste de todos os objetos que pertencem a um determinado conceito e uma intensão é um conjunto de atributos compartilhados por estes objetos. Normalmente, é uma tarefa difícil, senão impossível, enumerar todos os objetos e atributos de um determinado conceito. Assim, muitas vezes, os conjuntos de objetos e atributos são restritos a conjuntos discretos e finitos.

Definição 1 (Contexto). Um *contexto (formal)* é uma tripla $\mathfrak{t}=(G,M,I)$, onde G e M são conjuntos e $I \subseteq G \times M$. Os elementos de G e M são denominados **objetos** e **atributos**. Quando os conjuntos G e F são finitos, o contexto é dito **finito**.

A partir da noção de contexto, define-se a noção de conceito (formal):

Definição 2 (Conceito). Seja (G,M,I) um contexto. Para $A \subseteq G$ e $B \subseteq M$, definam-se os conjuntos:

$$A' = \{ m \in M \mid (\forall g \in A) \Rightarrow (g,m) \in I \} \quad B' = \{ g \in G \mid (\forall m \in B) \Rightarrow (g,m) \in I \}.$$

Um **conceito** num contexto $\mathfrak{t}=(G,M,I)$ é um par (A,B) , $A \subseteq G$ e $B \subseteq M$, tal que $A'=B$ e $B'=A$. Os conjuntos A e B são denominados, respectivamente, **extensão** e **intensão** do conceito. O conjunto de todos os conceitos de um contexto é denotado por $\mathfrak{b}(G,M,I)$.

Para tornar mais claras as definições anteriores, considere-se um contexto para aprendizado de conceitos sobre o sistema solar:

Tabela 1. Contexto adaptado de Davey e Priestley (2002) para aprendizado do sistema solar. Objetos são formados pelos planetas e atributos estão relacionados com observações astronômicas de tamanho, distância ao sol e presença de luas.

Planeta	Tamanho	Distância ao Sol	Possui lua ?
Mercúrio	pequeno	perto	não
Vênus	pequeno	perto	não
Terra	pequeno	perto	sim
Marte	pequeno	perto	sim
Júpiter	grande	longe	sim
Saturno	grande	longe	sim
Urano	médio	longe	sim
Netuno	médio	longe	sim
Plutão	médio	longe	sim

Um conceito bastante simples sobre o contexto anterior é formado pelos conjuntos:

$$A = \{ Terra , Marte \} \text{ e } B = \{ Tamanho-pequeno, Distância-perto, Lua-sim \}.$$

Este conceito caracteriza, de maneira bastante elementar, os objetos Terra e Marte através de três atributos bastante simples: tamanho pequeno, ficam perto do Sol e possuem lua. Assim, num sistema de aprendizado dentro do contexto dado, a resposta à questão “*Quais as características dos planetas Terra e Marte?*” poderia ser: “*São pequenos, estão perto do Sol e possuem lua.*”. Por outro lado, uma pergunta do tipo “*Quais planetas do Sistema Solar são pequenos, estão perto do Sol e possuem lua?*” poderia ter como resposta “*Terra e Marte.*”.

2.2 Reticulados Conceituais

É possível dotar o conjunto $\mathfrak{h}(G,M,I)$ de uma ordem parcial \leq através do seguinte resultado:

Teorema 1. *Sejam (A_1, B_1) e (A_2, B_2) dois conceitos em $\mathfrak{h}(G,M,I)$. Então a relação \leq definida por*

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$$

é uma relação de ordem parcial em $\mathfrak{h}(G,M,I)$. Ademais, $(\mathfrak{h}(G,M,I), \leq)$ é um reticulado completo.

A prova deste resultado é bastante simples e será omitida neste trabalho. É importante observar que a definição da relação de ordem parcial \leq dependeu somente do conjunto de objetos e não dos atributos. Porém, como $A_1 \subseteq A_2$, então tem-se que $A_2' \subseteq A_1'$, ou seja, que $B_2 \subseteq B_1$. Assim, é possível também caracterizar a ordem entre os conceitos através da relação $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow B_2 \subseteq B_1$. A escolha entre definir a ordem por objetos ou atributos é uma decisão do processo de implementação da estrutura de ordem e será discutida posteriormente.

Definição 3 (Reticulado Conceitual). Seja $\mathfrak{t}=(G,M,I)$ um contexto. Denomina-se *reticulado conceitual* do contexto \mathfrak{t} ao reticulado $(\mathfrak{h}(G,M,I), \leq)$.

Reticulados conceituais são melhor visualizados através dos Diagramas de Hasse, que permitem exibir propriedades de cobertura das ordens através de um diagrama hierárquico. Seriam necessários, teoricamente, dois Diagramas de Hasse para representar as duas definições de ordem possíveis: via objetos e atributos. Porém, é possível combinar estes dois diagramas em um único conforme mostrado em Davey e Priestley (2002).

A Figura 1 exibe os Diagramas de Hasse combinados, objetos e atributos associados aos reticulados conceituais da Tabela 1.

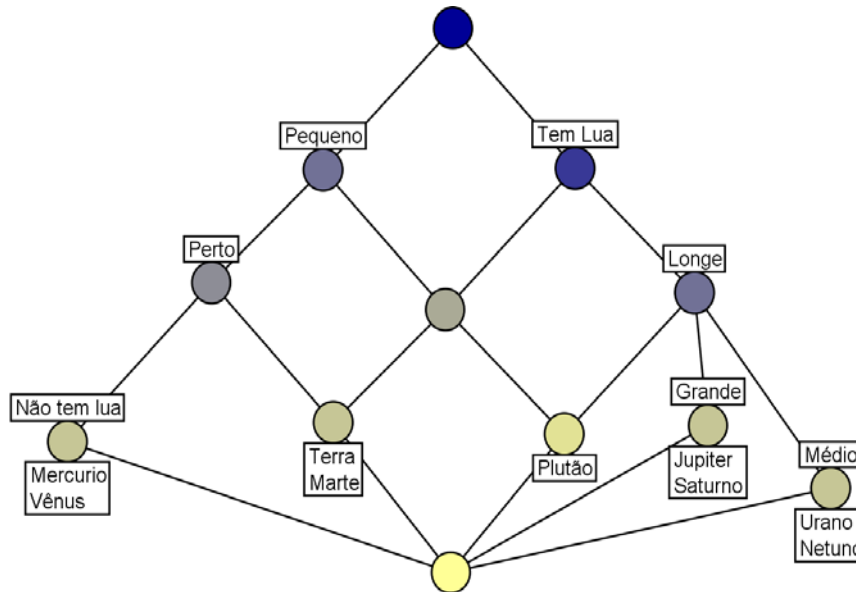


Fig. 1. Diagrama de Hasse combinado dos reticulados conceituais associados ao contexto de planetas do Sistema Solar. Anotações abaixo de cada nó representam conjuntos de objetos, enquanto que as acima representam atributos. A ausência de anotações inferiores e superiores pode significar ou que são facilmente calculadas através de hereditariedade ou que realmente não existem (conjunto vazio)

O diagrama combinado permite a navegação em duas direções verticais: ascendente e descendente. A navegação ascendente parte do ínfimo do reticulado e

permite obter facilmente os objetos. Por exemplo, no diagrama anterior o ínfimo não possui nenhum objeto vinculado (conjunto vazio), enquanto que todos os planetas aparecem no primeiro nível acima. O nó central do reticulado, apesar de estar sem anotação, herda os objetos dos níveis, respeitando-se a ordem de inclusão, imediatamente abaixo (Terra, Marte e Plutão). O supremo do reticulado na navegação ascendente, por hereditariedade, contém todos os objetos do contexto.

A navegação descendente permite obter facilmente os atributos. No ínfimo da navegação descendente, tem-se novamente um conjunto vazio de atributos, pois a Tabela 1 mostra que o conjunto de todos os objetos não possui todos os atributos em comum. Novamente, o nó central do reticulado não possui marcação de atributo, mas pode-se inferir facilmente, via hereditariedade, que ele é pequeno e possui lua. Claramente, o supremo na navegação descendente possui todos os atributos porque se tem um conjunto vazio de objetos.

Os nós do reticulado conceitual serão mapeados, posteriormente, para objetos de aprendizagem SCORM, que serão interligados através de arestas XLINK semanticamente anotadas. Cada nó receberá anotações de objetos e atributos. O reticulado conceitual como um todo representará um grande objeto SCORM.

A flexibilidade da ordem de navegação permitirá duas direções de navegação: uma ascendente (mais operacional), que privilegia os objetos, e outra descendente (mais axiomática), atrelada aos atributos. Em ambas as direções, as características hierárquica e hereditária da navegação estão fortemente presentes.

3 Objetos SCORM

O SCORM é um modelo de referência para o empacotamento e agregação de objetos de aprendizagem que possibilita sua utilização a partir de qualquer LMS (*Learning Management System*) compatível. O modelo SCORM foi apresentado pela ADL (*Advanced Distributed Learning*) e é definido através de três documentos: *Content Aggregation Model* (CAM), *Run-Time Environment* (RTE) e *Sequencing and Navigation* (SN) [ADL 2005].

As unidades básicas no modelo SCORM são os SCOs (*Sharable Content Objects*), os quais representam objetos de aprendizagem que fazem parte da estrutura de um curso. A navegação entre estes objetos deve ser definida para que possa ser seguida pelo LMS. Esta etapa da definição de um pacote SCORM é chamada de *agregação de conteúdo* e é realizada a partir da criação de um arquivo XML com as regras de navegação *entre* os objetos (nunca *intra-objetos*). Assim, um pacote SCORM deve conter um *arquivo de manifesto* (que segue as normas do IMS *Global Learning Consortium*) que declara o conteúdo do pacote, descreve a ordem de navegação do conteúdo (formado pelos SCOs) e indica onde os arquivos físicos que representam os SCOs podem ser encontrados.

Além do manifesto e dos arquivos físicos que implementam os SCOs, o pacote SCORM deve incluir arquivos que apresentam a descrição de cada SCO para facilitar sua manipulação. Estes são conhecidos como arquivos de *metadados*, já que contêm dados que descrevem outros dados. Basicamente, eles incluem informações tais como autor, título, versão, data de criação, requisitos técnicos, contexto educacional e objetivo.

O arquivo de manifesto pode referenciar outros, denominados sub-manifestos. O LMS utiliza estes arquivos para estabelecer a ordem de navegação entre os SCOs. Esta ordem é definida através da Árvore de Atividade (*Activity Tree*). Uma atividade de aprendizagem pode ser um recurso (atividade “folha”) ou pode ser composta de diferentes sub-atividades. Além disso, as atividades possuem início e fim definidos, assim como testes finais associados. A passagem de uma atividade para outra depende, além da seqüência correta, de tentativas (*attempts*) bem sucedidas nos testes finais.

A seqüência a ser seguida pelo LMS, portanto, é baseada no percurso de uma Árvore de Atividades derivada a partir do conteúdo do(s) arquivo(s) de manifesto. Esta estrutura, no entanto, nem sempre é a mais adequada para determinados cursos. Alguns trabalhos têm sido desenvolvidos com o objetivo de propor estruturas de seqüências alternativas.

Diversos autores vêm apontando limitações no modelo de metadados do SCORM. Abdullah et al. (2004) apontam que, apesar do padrão SCORM versão 2004 [ADL 2005] suportar o modelo de seqüenciamento proposto pela IMS [IMS 2004], tal modelo é demasiado simplista, de modo a não fornecer mecanismos para a implementação efetiva de objetos de aprendizagem adaptativos. Gomes et al. (2005) ressaltam as limitações que os atuais padrões, entre os quais o SCORM, possuem no sentido de representar adequadamente objetos de aprendizagem denominados funcionais, no sentido que sejam “artefatos computacionais cuja funcionalidade deve promover interação entre entidades”. Simões *et al.* (2004) propõem uma extensão ao SCORM de modo a suportar informações transversais aos objetos de aprendizagem, como regras de avaliação, currículo ou bibliografia. O artigo de Chang *et al.* (2005) apresenta uma proposta de uso de Redes de Petri para representar a árvore de atividades do modelo SCORM. O objetivo é permitir a visualização linear do fluxo percorrido, assim como possibilitar o “salto” de algumas lições arbitrariamente escolhidas. Os autores defendem que tal esquema pode ser útil em ambientes colaborativos (não cobertos pelo padrão SCORM).

4 Anotações e Links em SCOs

Conforme discutido na seção anterior, a flexibilidade de percurso entre SCOs é altamente dependente da estrutura da Árvore de Atividades. Esta seção apresenta uma proposta alternativa para implementação da Árvore de Atividades através da introdução de anotações de navegação XLink diretamente nos SCOs, eliminando a necessidade de uma estrutura maior para a árvore.

A Seção 2 apresentou uma estrutura de navegação hierárquica de objetos e atributos que transcende principalmente a dependência da navegação linear, permitindo a navegação segundo determinadas ordens. Para que os princípios apresentados naquela seção possam ser mapeados com mínimo impacto nas anotações de cada SCO, será adotada a seguinte ordem de construção: anotação de objetos e atributos, introdução de *links* entre objetos e anotações nos *links*.

4.1 Anotações de Objetos e Atributos

A anotação de objetos e atributos é realizada na seção META-DATA do arquivo de manifesto de um SCO, via marcador `keyword`, disponível no espaço de nomes `lom` [IEEE LTSC 2005] e descrito no Modelo de Agregação de Conteúdo SCORM (SCORM CAM).

A sintaxe geral de anotação é dada a seguir:

```
<manifest ... >
...
<metadata>
  <lom:lom xmlns:lom="http://ltsc.ieee.org/xsd/LOM">
    <lom:general>
      <lom:keyword >
        <lom:string> object | attribute </lom:string>
        <lom:string> Nome do objeto ou atributo </lom:string>
      </lom:keyword>
      ...
      <lom:keyword >
        <lom:string> object | attribute </lom:string>
        <lom:string> Nome do objeto ou atributo </lom:string>
      </lom:keyword>
    </lom:general>
  </lom:lom>
  ...
</metadata>
...
</manifest>
```

Cada SCO pode encapsular um conjunto de objetos e atributos. Cada objeto ou atributo utiliza uma *keyword*, descrita através de dois *strings*: tipo (*object* ou *attribute*) e nome do objeto/atributo. A diferenciação entre os dois tipos de anotação é importante para o cálculo de hereditariedade, que deverá ser realizado no processo de navegação.

4.2 Links entre SCOs e Anotações

A estrutura de navegação entre SCOs ainda é alvo de intensa discussão, conforme mostra o documento *SCORM Sequencing and Navigation* [ADL 2004]. Para o propósito de navegação no modelo hierárquico mostrado na Seção 2, a tecnologia XLink [Wilde e Lowe 2002] é extremamente adequada. *Links* especificados através da tecnologia XLink permitem navegação bidirecional, associação de regras de processamento do *link* e múltiplas direções.

Links serão adicionados como um novo marcador dentro do namespace lom, denominado `navigation`, conforme mostrado na sintaxe abaixo:

```
<manifest ... >
...
<metadata>
  <lom:lom xmlns:lom="http://ltsc.ieee.org/xsd/LOM">
    <lom:general>
      <lom:keyword >
        <lom:string> object | attribute </lom:string>
        <lom:string> Nome do objeto ou atributo </lom:string>
      </lom:keyword>
      ...
      <lom:keyword >
        <lom:string> object | attribute </lom:string>
        <lom:string> Nome do objeto ou atributo </lom:string>
      </lom:keyword>
    </lom:general>
    <lom:navigation xmlns:xlink="http://www.w3.org/1999/xlink"
      xlink:type = "extended"
      xlink:to = Endereço do próximo objeto SCORM
      xlink:from = Endereço do objeto SCORM anterior
      xlink:arcrole = Regra de processamento do link
      xlink:show = "replace"
      xlink:actuate = "onRequest"
    > Texto de navegação </lom:navigation>
    <lom:navigation ...>... </lom:navigation>
    ...
    <lom:navigation ...>... </lom:navigation>
  </lom:lom>
  ...
</metadata>
...
</manifest>
```

Cada marcador representa uma unidade básica de navegação: sabe-se de onde vem (`from`), para onde se vai (`to`) e qual a regra de processamento nestas navegações. Como sugestão, o formato de atuação no link é sob requisição (`onRequest`) e seu comportamento e deve ocorrer a substituição do conteúdo atual (`replace`).

Um dos detalhes mais importantes do *link* especificado consiste na regra de processamento: se a navegação for ascendente, os objetos válidos até o momento devem ser formados pelos objetos dos ancestrais mais os objetos do nó atual; se a navegação for descendente, os atributos válidos até o momento consistem também dos atributos dos ancestrais mais aqueles do nó atual. Além deste processamento canônico, é possível ainda verificar pré-requisitos na navegação, muito semelhantes aos marcadores do namespace imsss [IMSSS 2005].

A regra de processamento do *link* pode ser especificada indicando-se a referência de um processador (*parser*) XML baseado em SAX ou DOM para tratamento de XLink. Neste contexto, regras de tratamento do *link* são interpretadas como interesses ortogonais às relações de ordem – que podem vir a ser implementadas como aspectos no paradigma de Programação Orientada a Aspectos-AOP [Kiczales 1997] – e são mapeados como objetos externos ao SCO. Assim, pode-se encapsular dentro de cada SCO a referência externa –um ponto de junção, segundo a AOP – para um tratamento personalizado de cada *link*, permitindo alto grau semântico às relações de ordem estabelecidas no reticulados conceitual. Além disso, pode-se alterar a regra de tratamento sem necessidade de se reestruturar o SCO e liberando o LMS da tarefa de controlar a tarefa de navegação nos objetos SCORM.

A introdução do marcador `navigation` dentro do namespace `lom` pode ser feito de maneira bastante simples através da DTD abaixo:

```
<!ELEMENT navigation (#PCDATA)>
<!ATTLIST navigation
  xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink/namespace/
  xlink:type CDATA #FIXED "extended"
  xlink:to CDATA #REQUIRED
  xlink:from CDATA #REQUIRED
  xlink:arcrole CDATA #REQUIRED
  xlink:show (new|embed|replace) "replace"
  xlink:actuate (onRequest|auto) "onRequest"
>
```

Assim, a partir deste momento, cada SCO possui, além das suas características regulares, anotações de objetos e atributos, *links* bidirecionais para navegação ascendente e descendente, tudo encapsulado em uma única unidade representada através de seu arquivo de manifesto.

5 A Estrutura Hierárquica e o Meta-Objeto SCORM

As unidades SCORM construídas na seção anterior serão agora organizadas numa estrutura hierárquica para refletir a estrutura do reticulado conceitual. Nós do reticulado são mapeados para SCOs e relações de ordem através dos links especificado via XLink, cuja unidade total é representada através de um arquivo-manifesto. Cada arquivo-manifesto fará parte de uma rede iniciada por um objeto maior, denominado meta-objeto SCORM ou meta-SCO.

Tome-se como exemplo o Diagrama de Hasse combinado da Figura 1. A figura a seguir exibe os objetos SCORM associados ao nó-base (manifest0.xml), ao nó identificado pelos objetos Júpiter-Saturno e atributo Grande (manifest4.xml), e o nó que contém somente o atributo Longe (manifest8.xml).

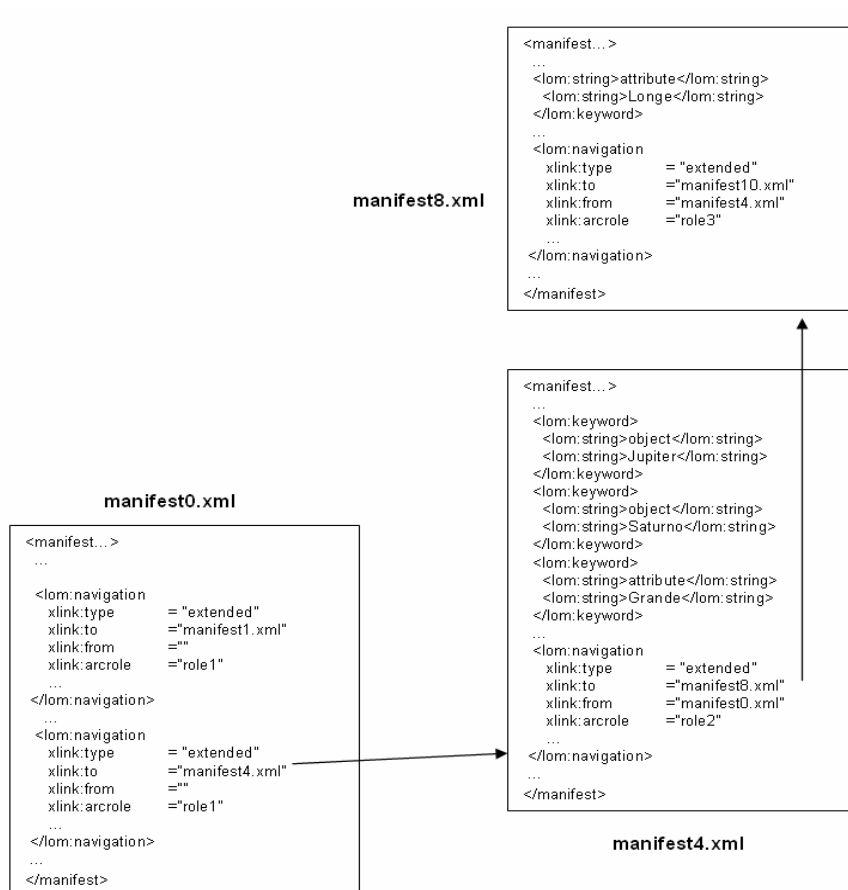


Fig. 2. Diagrama parcial de objetos SCORM (anotações e links), correspondente ao Diagrama de Hasse combinado na Figura 1. Conforme mostrado no Diagrama de Hasse, alguns objetos não possuem nem anotações de objetos nem de atributos (manifest0.xml), alguns somente de atributos (manifest8.xml), alguns só de objetos e outros possuem os dois (manifest4.xml). A omissão destas anotações representa uma grande economia de espaço para grandes reticulados e podem ser obtidas facilmente via regras associadas aos arcos de navegação.

A Figura 2 mostra que a estrutura hierárquica presente no diagrama conceitual pode ser facilmente mapeada para uma estrutura hierárquica de arquivos-manifesto. Numa navegação ascendente, isto é privilegiando os objetos, os atributos `xlink:to` e `xlink:from` são tratados de forma natural. Porém, na navegação descendente (ordem dos atributos), os atributos `xlink:to` e `xlink:from` devem ser processados de forma invertida.

Duas anotações são de vital importância para o meta-SCO: SCO inicial para navegação e sentido na navegação (ascendente ou descendente). Somente o SCO

inicial será mapeado como sub-manifesto do meta-SCO, permitindo reduzir o espaço necessário para armazenar a hierarquia dos outros SCOs.

O SCO inicial para navegação, bem como o sentido de navegação, podem ser facilmente introduzidos num arquivo de manifesto via marcador `<organization>` utilizado conjuntamente com o marcador `<resource>`, conforme mostrado abaixo para o exemplo da Figura 2:

```
<manifest>
  ...
  <organizations>
    <organization>
      <item identifier="Initial"
            Identifierref="RInitial"
            Parameters="?Order=Ascent">
      </item>
    </organization>
  <resources>
    <resource identifier="RInitial"
              href="manifest0.xml">
    </resource>
  </resources>
</manifest>
```

Um objeto inicial na hierarquia de navegação sempre será identificado através do nome `Initial`, colocado no parâmetro `identifier` do marcador `item`. O sentido da navegação é passado através do parâmetro `Order` para o arquivo `manifest0.xml`, especificado em `resource`. Caso a navegação seja ascendente, utiliza-se o valor `Ascent` e, para navegação descendente, `Descent`.

Assim, basta o LMS saber qual o SCO inicial para navegação e transferir-lhe o controle da navegação. Isto permite uma grande flexibilidade na navegação, porque pode-se associar comportamentos semânticos sofisticados na transição de um SCO para outro, além de permitir uma navegação bidirecional.

6 Conclusões e Trabalhos Futuros

A especificação de padrões de interoperabilidade, acessibilidade e reuso para objetos de aprendizagem é de grande interesse para Sistemas Gerenciadores de Aprendizado. Em particular, o padrão SCORM representa uma excelente alternativa para encapsular dados referentes a um objeto de aprendizagem.

Apesar de bastante geral, o padrão SCORM apresenta algumas deficiências, como o fato de carecer de mecanismos de navegação mais sofisticados. Este trabalho apresentou uma proposta de arquitetura de navegação em redes de objetos SCORM via reticulados conceituais. Estes reticulados permitem navegação em objetos e atributos de maneira bidirecional. A arquitetura é baseada na introdução de anotações e *links*, via tecnologia XLink, bastante difundida na integração entre documentos XML.

As anotações e *links* produzem um impacto muito baixo na atual estrutura do padrão SCORM e permitem a construção de redes complexas de objetos SCORM através de construções muito simples. *Links* entre objetos podem ser dotados de

processamento semântico altamente qualificado e permitem abstrair as ligações como aspectos entre arquivos-manifesto associados aos objetos de aprendizagem.

Trabalhos futuros incluem o estudo de modelos conceituais mais elaborados como, por exemplo, modelos híbridos envolvendo Redes de Petri e Reticulados Conceituais. Além disso, a geração automática de objetos SCORM via reticulados conceituais poderia ser uma poderosa ferramenta para auxiliar o desenvolvimento de objetos de aprendizagem com alta coesão, granulosidade fina e altamente adaptáveis a vários estilos de navegação.

Referências

Abdullah, N. A.; Bailey, C. e Davis, H. Synthetic hypertext and hyperfiction: Augmenting SCORM manifests with adaptive links. Em: *Proceedings of the 15th ACM Conference on Hypertext and Hypermedia*. Santa Cruz, EUA, 2004, pp. 183-184.

ADL. Advanced Distributed Learning. Disponível em: <http://www.adlnet.org>. Acesso em: 10.nov.2005.

Chang, W., Lin, H. W., Shih, T. e Yang, H. *SCORM Learning Sequence Modeling with Petri Nets in Cooperative Learning*. Em: *Learning Technology*, IEEE Computer Society, Volume 7, Issue 1, Janeiro de 2005. Disponível em: http://ltf.ieee.org/learn_tech/issues/january2005/#_Toc98675005. Acesso em 21.nov.2005.

DCMI, Dublin Core Metadata Initiative. *Dublin Core Metadata Element Set, Version 1.1: Reference Description*, 2004. Disponível em: <http://www.dublincore.org/documents/dces/>. Acesso em: 21.set.2005.

Davey, B.A., Priestley, H.A. *Introduction to Lattices and Order*. 2a. ed. Cambridge University Press, Cambridge, 2002.

Gomes, S. R.; Gadelha, B. F.; Mendonça, A. P. e Amortti, M. S. M. Objetos de Aprendizagem Funcionais e as Limitações dos Metadados Atuais. Em: *Anais do XVI Simpósio Brasileiro de Informática na Educação – SBIE 2005*. Juiz de Fora, Brasil, pp. 211-221.

IEEE LTSC. *Learning Object MetaData*. Disponível em: <http://ieeeltsc.org/wg12LOM/>. Acesso em: 20.set.2005.

IMS Global Learning Consortium. *Learning Resource Meta-data Specification - Version 1.2.4 – XSD Schema*, 2004. Disponível em: <http://www.imsglobal.org/metadata/index.html>. Acesso em: 21.set.2005.

IMSSS. IMS Global Learning Consortium. Disponível em: <http://www.imsglobal.org/xsd/imsss>. Acesso em: 25.set.2005.

Kiczales *et. al.* Aspect-Oriented Programming. Em: *Proceedings of the European Conference on Object-Oriented Programming (ECOOP'97)*, Springer-Verlag, 1997..

Simões, D.; Luís, R. e Horta, N. Enhancing the SCORM Metadata Model.. Em: *Proceedings of the 13th international World Wide Web Conference*. New York, EUA, 2004, pp. 238-239.

Wilde, E., Lowe, D. *XPath, XLink, XPointer and XML: A Practical Guide to Web Hyperlinking and Transclusion*. Pearson Education, 2002.

Wiley, D. A. *Learning Object Design and Sequenceing Theory*. Tese de Doutorado, Brigham Young University. Provo, EUA, 2000.