Panorama dos Editores de Atividades de Aprendizagem em IMS Learning Design

Aladir F. Silva Júnior^{1,2}, Leandro Roberto da Silva², Clovis Torres Fernandes¹

¹Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA) – Laboratório de Aprendizagem e Interação (LAI) - 12228-901 – São José dos Campos – SP – Brasil

²Coordenação de Informática – Instituto Federal de Goiás – Campus Jataí - 75804-020 - Jataí – GO – Brasil

aladir@{gmail.com}, leandroroberto.br@gmail.com, clovistf@uol.com.br

Abstract. This paper presents a survey of learning activity editors based on IMS Learning Design standard. We aim at covering the state-of-the-art of such editors. For each tool, the paper describes its main features and its contributions toward the modeling of studies, courses, lessons, among other learning resources. Beyond previous works and based on them, this paper presents recent advances on IMS Learning Design editors.

Resumo. Este trabalho, de caráter exploratório, retrata o estado da arte dos Editores de IMS Learning Design de acordo com uma classificação por níveis de autoria. Nesse sentido, foram apresentadas as principais características de cada ferramenta e a contribuição de cada uma para o contexto da modelagem de material de estudo, cursos, lições e outros artefatos. Além disso, neste artigo retrata-se o IME (Instructional Model Environment) como uma proposta de evolução dos atuais editores de IMS Learning Design.

1. Introdução

A partir do surgimento da especificação IMS *Learning Design* (IMSLD) em 2003, diversas ferramentas passaram a ser desenvolvidas para dar o suporte necessário ao uso e disseminação desta especificação, sendo as principais as ferramentas de edição (*i.e.* Editores *Learning Design* ou LD) e os executores LD (*i.e. players*). Neste estudo estão representados apenas os Editores LD, que permitem aos seus usuários (*e.g.* professores, tutores, projetistas) estruturarem uma lição ou um curso seguindo a especificação IMSLD.

Algumas dessa ferramentas, além de apresentar um editor LD genérico, apresentam editores específicos para determinados modelos instrucionais. Contudo, esses editores só podem ser usados para gerar atividade de aprendizagem no dado modelo instrucional (Recourse), ou a composição é limitada a apenas os editores disponíveis, pois a definição de novos editores é muito dificultada, limitando o seu uso a apenas pessoal com grande conhecimento de programação (Collage), ou, se é possível criar novos editores e a composição de editores um dentro do outro é possível, a tarefa de composição tem que ser finalizada de forma manual e exigindo retrabalho (LAMS, Open GLM Prolix).

DOI: 10.5753/CBIE.SBIE.2013.336 336

Cada artigo descrevendo um novo ambiente de editor LD usualmente apresenta sua visão do estado da arte bem como os aspectos em que ele é superior aos demais. Mas nenhum artigo foi encontrado na literatura apresentando os editores LD da literatura com base em uma classificação de ambientes de editores LD, em que se mostram a sua capacidade e flexibilidade de geração de editores LD, com base nos aspectos do parágrafo anterior. Ou seja, a comparação é pontual, nem sempre levando em consideração um espectro mais amplo da área de pesquisa.

O objetivo deste trabalho é mostrar versões mais recentes dos ambientes editores LD, com foco na evolução dos ambientes editores e geradores de editores LD, com base numa classificação por níveis de autoria (Silva Júnior e Fernandes, 2012). Neste artigo retrata-se ainda o IME (*Instructional Model Environment*), como uma proposta de evolução dos atuais editores de IMS *Learning Design*.

2. Contexto da Pesquisa

Nesta seção são apresentados alguns conceitos para o entendimento de termos referenciados durante o relato do panorama dos editores de LD.

Modelo Instrucional

Um modelo instrucional (MI) é um conjunto de estratégias instrucionais, que podem ser derivados a partir de teorias pedagógicas ou de práticas bem sucedidas. As estratégias instrucionais são táticas utilizadas para a efetivação da aprendizagem. São as técnicas que o instrutor ou o projetista instrucional usa para engajar o aprendiz e facilitar a aprendizagem (Dabbagh, 2005). Para entendimento pode-se assemelhar o MI ao "modo de fazer" de uma receita culinária, onde os ingredientes são os elementos envolvidos no processo (e.g. papeis, recursos, ambiente) e o "modo de fazer" é o conjunto de estratégias para se realizar com sucesso a tarefa (i.e. método).

A Especificação IMS Learning Design

A IMS *Learning Design* (IMSLD) é uma especificação padrão que permite modelar o processo de aprendizagem e estruturar o conteúdo inerente a este processo e é mantida pelo IMS *Global Learning Consortium* (IMSLD, 2013). A especificação IMSLD pode ser considerada pedagogicamente neutra, pois sua notação abstrata é suficientemente genérica para representar, de maneira formal, as estruturas comuns encontradas em diversos modelos pedagógicos (KOPER & OLIVIER, 2004; KOPER e TATTERSALL, 2010).

A especificação IMSLD tem como produto final a geração de uma Unidade de Aprendizagem (UA), originalmente intitulada *Unit of Learning* (UoL), uma estrutura autocontida de recursos e a orquestração destes. Esse conceito é equivalente ao tratado como atividade de aprendizagem. A partir deste ponto usaremos apenas a sigla UoL para referenciar tal conceito que, conforme Koper e Olivier (2004, p.2) "pode ser um curso, um *workshop*, uma aula etc. que pode ser instanciada e reusada diversas vezes, de diferentes formas, por diferentes pessoas em um ambiente *on-line*".

A estrutura conceitual da IMSLD é formada em níveis sobrepostos, a saber: A, B e C. O nível A possui os elementos básicos (*i.e.* atividade, papel e ambiente, entre outros) que estruturam o IMSLD. O nível B permite a adição de propriedades e condições e o nível C acrescenta o conceito de notificação. Com uma UoL de nível A, a

execução segue o mesmo fluxo da criação das atividades. Com os níveis B e C as atividades especificadas ganham recursos adicionais que possibilitam o controle destas por meio de regras e, também, o monitoramento da execução da UoL, por meio de propriedades (IMSLD, 2013).

Editores IMS Learning Design

Os Editores IMS *Learning Design*, ou simplesmente Editores LD, são softwares que permitem aos seus usuários estruturarem uma lição ou um curso seguindo a especificação IMSLD. Como resultado, o Editor LD gera um arquivo ou conjunto de arquivos empacotados que será reconhecido e consumido pelo executor LD (*i.e. player* LD), que tornará acessível o conteúdo aos usuários finais de acordo com a estruturação feita no editor.

3. Metodologia e Delimitação da Pesquisa

Este estudo não pretende ser uma investigação aprofundada e em larga escala, contemplando todos os editores LD disponíveis. Contudo, foi realizada uma extensa pesquisa na Internet buscando literatura especializada e confiável, que resultou numa lista com os principais editores LD da literatura. Fazem parte dessa literatura artigos diversos, relatórios técnicos, sítios dos projetos que desenvolveram os editores LD e teses.

Num segundo momento, procedeu-se ao *download* dos editores e sua respectiva instalação. Posteriormente, procedeu-se ao uso de cada editor de LD fazendo uma avaliação em termos de funcionalidades e modo de trabalho, além do material final gerado (*i.e.* UoL pronta para execução). Por fim, efetuou-se a classificação do editor em relação aos níveis de autoria.

Entende-se que cada ferramenta de autoria (editores) tem sua especificidade e seria uma tarefa complexa classificá-las considerando todas suas nuances. No entanto, para este panorama foi usada a proposta de Classificação por Níveis de Autoria (Silva Júnior e Fernandes, 2012), que se julgou mais adequada principalmente quanto à evolução necessária de tais editores e pela exposição explícita da participação de autores na criação de UoLs. Essa classificação estabelece 6 níveis (0 a 5) e cada nível será explicado nas próximas seções, exceto o nível 0 que não se aplica aqui.

Foram descartados os editores que, apesar de estarem relatados na literatura não foram localizados para *download*. Um exemplo é o Editor COSMOS LD (Miao, 2005).

4. Editores de Atividades de Aprendizagem em IMS Learning Design

As ferramentas de autoria em IMSLD, encontradas na literatura e que atendem aos critérios que delimitaram esta pesquisa, são apresentadas nas próximas subseções.

4.1. Nível 1 – Editores Genéricos de IMSLD

Esta classe agrupou a maioria dos editores LD encontrados na literatura. Figuram aqui os editores LD que são genéricos, ou seja, atendem a um propósito geral na produção de UoLs. Possuindo ou não interface gráfica com o usuário, aplicando ou não a metáfora do teatro usada na especificação IMSLD, usando ou não a estrutura de árvore da IMSLD, as ferramentas deste nível permitem ao usuário a liberdade de estruturar seu

material do modo que melhor lhe convier, usando qualquer MI. Esta liberdade traz consigo a necessidade do usuário ter conhecimentos acerca do modelo instrucional que ele deseja usar para aquele material.

Com o uso de editores genéricos de IMSLD, é possível gerar material instrucional para um curso, um módulo de curso, um *workshop* ou uma apresentação, dentre outras possibilidades. A ferramenta desse nível não se preocupa com a qualidade do conteúdo, com o modelo instrucional aplicado, nem como será feita a dosagem desse conteúdo aos aprendizes, ficando essas responsabilidades a cargo do autor da atividade. São exemplos dessas ferramentas o CADMOS, o CopperAuthor, Editor LD LAI (Fernandes et al., 2012), LD *Authoring Tool*, MOT+LD, *Pathway* ASK-LDT e *Reload* (Reload, 2012).

4.2. Nível 2 – Editores Específicos de IMSLD

Esta classe agrupa editores voltados a um MI específico, seguindo uma abordagem pedagógica definida, ou seja, permite ao usuário estruturar material somente a partir de um determinado MI. Um exemplo de editor LD dessa classe é o *WQE Editor* que permite desenvolver UoLs seguindo a técnica de *WebQuest* (Camargo *et al.*, 2010). Foi desenvolvido no Laboratório de Aprendizagem e Interação (LAI) do Instituto Tecnológico de Aeronáutica (ITA) como uma extensão do Editor LD LAI, herdando assim as características de interface gráfica, formato de arquivos e tecnologias de desenvolvimento.

O WQE Editor segue as estratégias que compõem um MI baseado na técnica *WebQuest* (*e.g.* introdução, tarefa, processo, avaliação, conclusão) para conduzir o usuário por este MI sem a necessidade de conhecimento desta técnica, nem da sintaxe da especificação IMSLD. Por ser um editor de LD que atende especificamente a um MI, o WQE não permite o desenvolvimento de UoLs usando outras abordagens.

4.3. Nível 3 – Editores LD com Múltiplos Modelos Instrucionais

São editores LD desta classe aqueles em que, apesar de ter a disposição do usuário vários MIs, não é possível usar mais de um MI para cada UoL. Assim, o usuário pode estruturar seu material seguindo o apoio de um MI específico, como acontece com os editores LD específicos. A vantagem é que não é preciso ter uma ferramenta para cada modelo instrucional. Como fator limitante se tem a impossibilidade de integrar os diversos MIs disponíveis para estruturar uma UoL.

O *ReCourse*, um editor desenvolvido no âmbito do projeto europeu TENCompetence pela Universidade Bolton, é uma ferramenta representante dessa classe. É uma ferramenta *stand-alone* multiplataforma, desenvolvida em linguagem Java usando plataforma *Eclipse*. Uma cópia está disponível no sítio do projeto. O *ReCourse* apóia a especificação IMSLD em seus três níveis, a saber, A, B e C. Também suporta a importação e exportação de pacotes IMSLD.

O *ReCourse* apresenta um recurso de *templates* que são guias para o usuário seguindo um determinado modelo instrucional (MI). Isso se constitui em um diferencial, dada a possibilidade de um professor querer modelar seu material (*e.g.* curso, apresentação) e não precisar conhecer como fazer isso a partir de alguma orientação pedagógica já experimentada (*i.e.* padrões de aprendizagem baseados ou não em teorias

pedagógicas). O *ReCourse* oferece ao usuário a possibilidade de também criar seu *template*; ou seja, é uma forma de o usuário ter um editor LD específico para o seu modelo instrucional pessoal.

4.4. Nível 4 – Editores LD com Integração de Múltiplos Modelos Instrucionais

Essa classe é uma evolução em relação ao discutido na Seção 4.3., pois agrupa ferramentas que permitem ao usuário usar mais de um MI para estruturar cada UoL, até então uma limitação no processo de autoria. Um recurso importante dessa classe de editores LD é a possibilidade de reúso dos MIs, seja por meio da combinação entre eles, seja pelo uso destes para estruturar um material didático. São exemplos de editores LD desta classe o *Collage*, o LAMS e o *Open* GLM *Prolix*.

Collage é um editor LD que utiliza o conceito de padrões para fluxos de aprendizagem colaborativos, em inglês, Collaborative Learning Flow Patterns (CLFP). É uma aplicação stand-alone e de código-fonte aberto¹. Foi desenvolvido na universidade de Valladolid, na Espanha, a partir da implementação do Reload Editor.

Para estruturar o material instrucional, o usuário é apresentado a alguns CLFPs (e.g. *pyramid e jigsaw*), que correspondente ao que se aborda aqui como modelo instrucional (MI), e pode escolher um para a estruturação de seu material de aprendizagem. Para permitir o uso de mais de um MI, o Collage dá a possibilidade do usuário estabelecer a quantidade de fases para o MI inicial e depois substituir alguma dessa(s) fase(s) por um outro MI. Após a escolha do padrão de fluxo, o professor recebe apoio na criação de UoLs baseadas naquele CLFP escolhido que são gravadas no formato IMSLD, permitindo qualquer executor LD as utilizar (Jorrín-Abellán *et. al.* 2006).

Outra ferramenta desta classe é o LAMS, um sistema baseado em *Web*, desenvolvido no *E-learning Centre of Excellence* (MELCOE) na *Macquarie University*. Foi desenvolvido em Macromedia Flash e uma cópia está disponível no sítio do projeto². Dá suporte ao IMSLD no nível A. Permite também exportar/importar pacotes IMSLD a partir da versão de 2008. O LAMS trabalha em uma estrutura baseada em elementos gráficos que representam objetos da IMSLD.

O *Open* GLM *Prolix* é um aplicativo multiplataforma, *stand-alone*, baseado no *Reload* LD *Editor*. Foi desenvolvido usando linguagem Java e plataforma Eclipse, no âmbito do projeto europeu Prolix da Universidade de Viena, Áustria, e tem código-fonte aberto disponível para *download*³. Este editor LD suporta a IMSLD nos níveis A e B e, apesar de permitir a exportação de pacotes, não é possível importar pacotes IMSLD.

O *Open* GLM *Prolix* em sua estrutura de interface com o usuário sugere a montagem de um fluxo de aprendizagem por meio de diagramas com objetos e setas para conexão entre tais objetos. Esta organização em formato de diagrama de fluxo, com indicador de início e fim, assemelha-se uma especificação de MI, pois é possível salvar o MI e inseri-lo em outro MI. Isso é feito abrindo o MI a ser inserido (*i.e.* origem) em outra aba e selecionando os seus elementos, copiando e colando no MI destino. A

¹ http://www.gsic.uva.es/collage/download

² http://lamsfoundation.org/downloads_home.htm. Acesso em 30 jul. 2013.

³ http://sourceforge.net/projects/openglm - Acesso em 01 ago. 2013.

reorganização no MI destino é feita manualmente, ajustando os novos elementos e refazendo o indicador de início e fim quando necessário, bem como ajustando as conexões para apontar para os elementos do novo MI que está sendo inserido.

Os editores *Collage*, LAMS e *Open* GLM *Prolix* estão na direção de permitir que o usuário possa organizar seu material instrucional no formato da IMSLD, mas também de dar flexibilidade nesse processo, permitindo ao professor adequar esse material de acordo com suas pretensões e também reusar estruturas desenvolvidas em outro momento ou por outros professores.

Algumas limitações foram notadas, no *Collage*: (i) São disponibilizados para uso e intercalação (composição) apenas seis CLFPs; (ii) Caso o professor queira criar seu próprio modelo instrucional (CLFP, no caso) que vá além da combinação dos seis modelos disponíveis, isso se torna muito trabalhoso e depende de conhecimentos não triviais de programação de computadores, especialmente na linguagem Java.

No LAMS e no Open GLM Prolix a limitação se refere à necessidade do usuário ter que reorganizar manualmente o MI quando da inserção de outro MI, ajustando os elementos inseridos bem como as ligações entre eles. Não é possível substituir uma fase de um MI com outro MI. É necessário que o usuário tenha que repensar todo o conjunto novamente.

5. Nível 5 – Evolução do Estado da Arte

Apresenta-se nesta seção a visão geral de um modelo que busca contribuir com a construção de editores LD no Nível 5, nível mais alto da classificação usada neste trabalho, em que os problemas dos editores do nível 4 foram eliminados. Foi desenvolvido e está na fase de testes e experimentação, no âmbito de pesquisa em andamento dos autores desse artigo, uma prova de conceito: um ambiente denominado *Instructional Model Environment* (IME) que trabalha diretamente com a estrutura de árvore do IMSLD e tem demonstrado atingir os objetivos do Nível 5 da Classificação por Níveis de Autoria.

Seguindo um modelo conceitual (*vide* Figura 1) que usa o padrão de empacotamento IMS *Content Packaging* (IMSCP), uma Linguagem para Modelos Instrucionais (LMI) e um conversor de LMI→IMSLD e IMSLD→LMI, o IME permite aos seus usuários: (i) criar seus próprios MIs, (ii) criar UoLs com o suporte do MI especificado, (iii) compor MI a partir de outro(s) MI dentro da mesma estrutura em árvore (iv) exportar no formato de pacotes do IMSLD (v) importar pacotes IMSLD.

O ambiente IME é composto por quatro elementos que, ao mesmo tempo representam os estágios para uso e se constituem em módulos funcionais da ferramenta (*i.e.* construção, edição, execução e distribuição). A cada um desses elementos ou fases são atribuídos papeis distintos de usuários. Na fase 1 (*i.e.* construção), por exemplo, o usuário deverá ser um especialista-autor em MI, pois nesta fase é permitido ao usuário criar um novo Modelo Instrucional (MI), semelhantemente ao que é feito pelo recurso de *templates* implementado na ferramenta *Recourse*.

Na fase 2 (*i.e.* edição) o usuário será um professor-autor que poderá compor um MI a partir de outro MI além de enriquecer a modelagem com inserção de Objetos de Aprendizagem, dando dicas, por exemplo para um usuário leigo que irá usar o MI enriquecido na terceira fase.

Na fase 3 (*i.e.* execução) o usuário será um professor-usuário, que necessitará apenas de conhecimento em sua própria área de atuação ou do material que ele quer modelar na ferramenta. A execução a que se refere a fase 3 não corresponde à disponibilização das UoLs

desenvolvidas em executores LD (função da fase 4), mas sim executar um editor com o MI enriquecido na fase 3! O ambiente disponibiliza um editor para aquele MI específico.

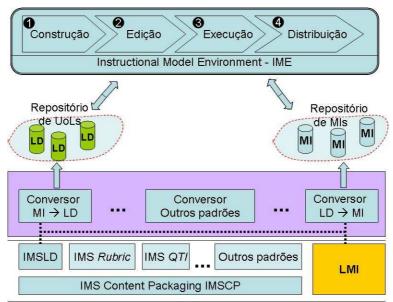


Figura 1 - Modelo Conceitual do IME.

Por fim, a fase 4 (*i.e.* distribuição) poderá ser feita por quaisquer usuários, pois consiste em transformar via conversor LMI→LD o material já modelado, empacotando-o no formato do padrão IMSCP e disponibilizando para qualquer executor LD.

Percebe-se que, nos testes iniciais, a ferramenta desenvolvida a partir do modelo IME, apesar de trabalhar no momento apenas no nível A da IMSLD, consegue transpor algumas barreiras impostas por esta especificação, possibilitando superar as limitações dos editores *Collage*, LAMS e *Open* GLM *Prolix* já apresentadas. Além disso, percebe-se que ao usar a estrutura de árvore do LD, a ferramenta tende a ser mais funcional e prática. A proposta de se trabalhar em fases, que podem ser associadas a usuários ou grupos de usuário específicos, permite diminuir uma possível relutância dos professores em dividir seu trabalho com especialistas (*e.g.* programadores, projetistas instrucionais e teóricos em educação, dentre outros).

O modelo ainda permite sanar algumas limitações da especificação IMSLD apontadas por Burgos (Burgos, 2008). Isso é possível usando uma linguagem intermediária, Linguagem de Modelos Instrucionais (LMI), que evita qualquer alteração na sintaxe da IMSLD, permitindo a disponibilização do material didático estruturado usando os mesmos executores (*i.e. players* LD) já disponíveis no mercado e literatura.

A possibilidade da inclusão de um MI dentro de outro MI traz ao professor mais flexibilidade na estruturação de seu curso, material didático, e também dá mais abrangência à IMSLD sem, contudo, alterar a especificação nem tampouco os executores LD. O professor poderá, à seu modo, organizar um material já modelado, reusando-o apenas com o uso de um recurso arrastar e soltar na mesma estrutura de árvore, como pode ser visto no recorte de tela exibido na Figura 2. No primeiro quadro da imagem é possível visualizar dois MIs (i.e. MI - Básico e MI - WebQuest). No segundo quadro visualiza-se em destaque o MI - WebQuest, onde o professor resolveu utilizar na etapa Tarefa o MI - Básico, concluindo assim uma composição do MI - WebQuest.

6. Resumo Comparativo de Características dos Editores LD

Para apresentar uma visão geral dos editores LD, apresenta-se o Quadro 1, onde foram consideradas algumas características principais para permitir comparação entre editores LD. A coluna intitulada "DAD" é uma abreviatura do inglês *drag and drop*, recurso "arrastar e soltar". A coluna intitulada "Nível" explicita a classificação em níveis de autoria (Silva Júnior e Fernandes, 2012), que se reflete na estrutura geral do quadro.

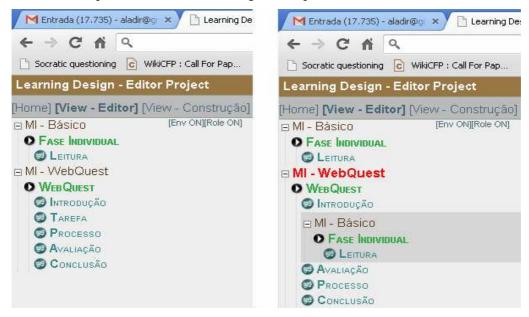


Figura 2 - Composição de Modelos Instrucionais no IME.

Quadro 1 - Resumo comparativo dos editores LD.

Editor IMSLD	GUI	Apresentação	DAD	Nível	Licença	Nível IMSLD	Importa IMSLD	Exporta IMSLD
		_	√	_				o
CADMOS		Diagrama	·	1	Freeware	A, B	Não	
CopperAuthor	✓	Árvore / Tabela	Não	1	Open Source	А	Não	✓
Editor LD LAI	✓	Formulário / Diagrama	✓	1	Open Source	Α	Não	✓
LD Authoring Tool	✓	Árvore / Botões / Formulário	√	1	Open Source	А	Não	✓
MOCOLADE	✓	Diagrama	✓	1	Proprietária	Α	Não	✓
MOT+LD	✓	Diagrama	✓	1	Proprietária	А	Não	✓
Pathway ASK-LDT	✓	Diagrama	✓	1	Freeware	A, B	Não	✓
RELOAD	✓	Formulário	✓	1	Open Source	A, B, C	Não	✓
WQE Editor	✓	Botões / Formulário	✓	2	Open Source	A	Não	✓
Recourse	✓	Formulário	✓	3	Open Source	A, B, C	✓	✓
Collage	✓	Diagrama	✓	4	Open Source	Α	Não	✓
LAMS	✓	Diagrama	✓	4	Open Source	А	✓	✓
Open GLM Prolix	✓	Diagrama	✓	4	Open Source	A, B	Não	✓
IME	✓	Árvore / Formulário	✓	5	Open Source	А	✓	√

7. Perspectivas de Pesquisa Futura

A partir do panorama apresentado, incluindo a proposta para evolução do estado da arte dos Editores LD, as seguintes possibilidades de pesquisa são vislumbradas:

- A interface com o usuário do editor WebQuest segue o padrão de interface usado no IME, empregado uniformemente em qualquer editor gerado por ele. Um trabalho futuro seria desacoplar a interface com o usuário para um dado editor LD no IME, de forma que seja possível definir uma nova interface com o usuário que seja mais apropriada no sentido de usabilidade ao tipo de editor gerado. Talvez seja preciso ter um editor de interfaces gráficas com o usuário para realizar isso.
- Desenvolver e verificar a LMI para o nível B e C, ampliando as possibilidades ofertadas pela IMSLD, por exemplo, em termos de condições simples, aritmética básica, configuração de propriedades, modo de visibilidade e camadas DIV.
- Usar a mesma técnica usada para criar o IME na criação de editores para outros padrões IMS, tais como Rubric e QTI, bem como para padrões fora do escopo do IMS, como o SCORM.
- Integrar outros editores de outros padrões ao IME, de modo a permitir aos usuários mais flexibilidade, sem expor o usuário às complexidades principalmente em termos de programação de tais padrões. Assim, poderíamos ter editores LD, Rubric, QTI e SCORM, com a possibilidade de misturar e compor editores multipadrão. Por exemplo, para editar um WebQuest, poderíamos criar um editor WebQuest que mistura as especificações IMSLD e Rubric; o editor permitiria criar uma webquest quase inteira no padrão IMSLD e fazer uso da especificação Rubric na parte de avaliação da webquest, que poderia corresponder a uma rubrica. Mas isso ficaria transparente ao autor: para ele é como se a edição fosse uma só; ele nem perceberia que outro editor foi acionado nesse ponto.
- Dado que os editores multipadrões se tornarão realidade, será necessário criar players (executores) multipadrões; por exemplo, um que entendesse IMSLD e Rubric, para poder executar uma atividade criada pelo editor LD/Rubrica do item anterior.

8. Considerações Finais

Este artigo apresentou um panorama dos editores LD, baseado em um estudo exploratório dessas ferramentas indispensáveis para a estruturação de material instrucional seguindo a especificação IMSLD.

Este artigo se antecipou ao que se tem relatado na literatura, apresentando um modelo/ferramenta que contorna problemas apresentados pelos atuais editores de IMSLD. Ou seja, com o ambiente IME é possível gerar atividade de aprendizagem num dado modelo instrucional; criação e composição de editores é ilimitada, pois pode-se criar um editor novo com base em um MI qualquer e pode-se fazer a composição de editores um dentro do outro de forma automática, sem precisar de ajuda manual do autor.

Agradecimento — Este trabalho decorre do trabalho sobre geradores de editores no padrão IMS Learning Design desenvolvido no escopo do projeto FAPESP – TIDIA/Ae Aprendizado Eletrônico, processo #05/50621-2.

9. Referências Bibliográficas

- BURGOS, D. What is wrong with the IMS Learning Design specification? Constraints And Recommendations. 2008. Disponível em http://www.kde.cs.uni-kassel.de/conf/lwa10/papers/abis8.pdf.
- CAMARGO, E. Z.; FERNANDES, C. T. *WQE um Editor de WebQuests Versátil.* Anais do XXI Simpósio Brasileiro de Informática na Educação, 2010. João Pessoa-PB.
- COLLAGE. *COLLAGE Collaborative Learning Design Editor*. Disponível em: http://www.gsic.uva.es/collage/>. Acesso em: 20/07/2013.
- DABBAGH, N. *Pedagogical models for E-Learning: A theory-based design framework.* International Journal of Technology in Teaching and Learning, p. 25–44, 2005.
- FERNANDES, C. T.; SILVA JÚNIOR, A. F.; VARCHAVSKY, M. Editor LD: Uma Ferramenta de Apoio à Autoria de Atividades de Aprendizagem no Padrão IMS Learning Design. II APPLETS APplications to Provide LEarning and Teaching Support. Rio de Janeiro-RJ, 2012.
- GRIFFITHS, D.; BLAT, J.; GARCIA, R.; VOGTEN, H.; KWONG, K. *Learning Design Tools*. Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. p.109–135, 2005. Netherlands: Springer-Verlag Berlin / Heidelberg.
- IMSLD. IMS *Global Learning Consortium Learning Design Specification*. Disponível em: http://www.imsglobal.org/learningdesign/>. Acesso em: 30/07/2013.
- JORRÍN-ABELLÁN, I.; RUIZ-REQUIES, I.; HERNÁNDEZ LEO, D.; VILLASCLARAS-FERNÁNDEZ, E. D.; DIMITRIADIS, Y.; RUBIA-AVI, B.; ASENSIO-PÉREZ, J. I. *Collage, a Collaborative Learning Design Editor Based on Patterns*. 2006. Disponível em http://hdl.handle.net/10230/5939. Acesso em 04 ago. 2013.
- KOPER, R.; OLIVIER, B. *Representing the learning design of units of learning*. Educational Technology & Society, p. 97–111, 2004. Netherlands.
- KOPER, R.; TATTERSALL, C. Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. 2010.
- MIAO, Y. CoSMoS: Facilitating Learning Designers to Author Units of Learning Using IMS LD. in LOOI, C., Jonassen, D., IKEDA, M. (Ed.): Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences, Proceedings of the 13th International Conference on Computers in Education. 2005. p. 275-282.
- RELOAD. RELOAD Reusable eLearning Object Authoring & Delivery. , Jun. 2012. Disponível em: http://www.reload.ac.uk/editor.html>. Acesso em: 03/8/2013.
- SILVA JÚNIOR, A. F. DA; FERNANDES, C. T. Autoria em IMS Learning Design: uma proposta de classificação por níveis. Anais do XXIII Simpósio Brasileiro de Informática na Educação, 2012. Rio de Janeiro RJ.