

OBJETOS DE APRENDIZAGEM: UMA PROPOSTA TECNOLÓGICA PARA FACILITAR A REUTILIZAÇÃO, ORGANIZAÇÃO E COMPARTILHAMENTO

Maio de 2007

Érico Galdino Almeida – Senac São Paulo – erico.galmeida@sp.senac.br

Leandro Cassa – Senac São Paulo – leandro.cassa@sp.senac.br

Silvia Fernanda Corrêa – Senac São Paulo – silvia.fecorrea@sp.senac.br

Categoria (Métodos e Tecnologias)

Setor Educacional (Educação Corporativa)

Natureza (Descrição de Projeto em Andamento)

Classe (Experiência Inovadora)

RESUMO

Este artigo tem como objetivo discutir uma das principais características dos objetos de aprendizagem: sua reutilização. É apresentada uma proposta tecnológica e metodológica para o desenvolvimento de objetos de aprendizagem reutilizáveis com o uso da tecnologia Flash, visando aumentar tanto a produtividade quanto a qualidade no desenvolvimento destes objetos.

Palavras-chave: objetos de aprendizagem; reutilização; e-learning; flash; scorm.

1- Introdução

A discussão acerca do uso de objetos de aprendizagem reutilizáveis - RLOs (*Reusable Learning Objects*), em especial sua implicação na dinâmica das instituições provedoras de conteúdo, não é nova. Apesar da resistência de muitos profissionais e instituições em adaptar sua metodologia de trabalho para produzir conteúdos reutilizáveis, a prática e a experiência de mercado mostram que não há como ignorar essa tendência, que veio pra ficar. Se na teoria reutilizar objetos parece uma coisa simples, na prática são encontradas várias barreiras, desde como conceber pedagogicamente esses objetos para que possam ser “desvinculados” de contexto, e, conseqüentemente, possam ser reaproveitados, até sua implementação tecnológica.

Frente a essas barreiras, o Núcleo de Educação a Distância do Senac São Paulo tem buscado adaptar sua metodologia de desenvolvimento de conteúdos e desenvolveu uma nova proposta tecnológica que traz em seu cerne a preocupação não só com o ganho da produtividade no desenvolvimento dos conteúdos, mas também abre a discussão para novas propostas de produção e troca coletivas de objetos de aprendizagem.

Neste artigo será explicitado como surgiu a idéia de desenvolver uma nova proposta de objetos de aprendizagem reutilizáveis e como isso se deu na prática. Além disso, busca incitar a discussão e troca de idéias com a comunidade de EaD para aprimorar a proposta aqui apresentada.

O foco deste trabalho está voltado àqueles que já possuem alguma familiaridade com o tema. Por esse motivo, não serão aprofundadas as explicações acerca de alguns conceitos-chave, como o que são objetos de aprendizagem, LMS e padrão SCORM.

2- Objetos de Aprendizagem Reutilizáveis

Atualmente, tanto os desenvolvedores de conteúdo e de ferramentas de gestão de alunos e cursos quanto os profissionais de educação envolvidos com a produção de materiais on-line reconhecem, ainda que em diferentes graus, as vantagens e ganhos em se utilizar objetos de aprendizagem.

Mas embora sejam cada vez mais adotados, não há uma definição universalmente aceita de objetos de aprendizagem. Há, sim, diversas definições de diferentes autores e organizações [1]:

Para o IEEE (*Institute of Electrical and Electronics Engineers*), objetos de aprendizagem “são definidos como qualquer entidade, digital ou não-digital, que podem ser usadas, reutilizadas ou referenciadas durante o processo de aprendizagem provido por tecnologias”. [2]

Objeto de Aprendizagem para BARRITT [1] “é uma coleção independente de conteúdos e elementos de mídia, uma abordagem de aprendizagem (interatividade, arquitetura de aprendizagem, contexto) e metadados (usado para armazenamento e busca)”.

Já para POLSANI [3], “objeto de aprendizagem é uma unidade auto-suficiente de conteúdo de aprendizagem que é predisposta para o reuso em múltiplos contextos instrucionais”.

Apesar das diferentes definições, há uma certa concordância que os objetos de aprendizagem devem ter alguns requisitos funcionais principais [3]:

- **Acessibilidade:** os objetos devem conter metadados para que possam ser armazenados e referenciados em bancos de dados.
- **Reutilização:** deve ser possível usar o objeto em diferentes contextos instrucionais, sem a necessidade de manutenção.
- **Interoperabilidade:** o objeto deve poder ser usado em diferentes plataformas de e-Learning.

Diante deste cenário, o Núcleo de Educação a Distância do Senac São Paulo adotou, para guiar seu processo de desenvolvimento, a seguinte definição:

É considerado objeto de aprendizagem qualquer material digital de aprendizagem com começo, meio e fim ou que tenha um fim em si mesmo.

O importante é que cada instituição adote a definição de objeto de aprendizagem que melhor se adeque a sua realidade, mas além da definição, é fundamental pensar a questão da granularidade, ou seja, qual o tamanho do menor item de conteúdo que será considerado um objeto de aprendizagem.

Nos principais projetos desenvolvidos pelo Senac, a seguinte estrutura foi utilizada (embora a proposta apresentada neste trabalho tenha se baseado na estrutura abaixo, ela não está restrita ou atrelada a este formato):

- Curso (contém diversos módulos/disciplinas)
 - Módulo/Disciplina (contem várias aulas)
 - **Aula** (contém vários Assets) → **Objetos de aprendizagem**
 - Asset (texto, figura, ilustração, animação, atividade etc.)

Entre as principais dúvidas sobre a real possibilidade de reutilização de conteúdos que surgem em discussões sobre objetos de aprendizagem, sempre aparecem questões sobre layout, cores, tipo de navegação, uso e controles do SCORM, além de outros pontos similares.

Por esse motivo, este trabalho busca uma solução que facilite a produção de objetos e traga ganhos significativos de produtividade e qualidade no seu desenvolvimento.

3- Padrão SCORM

Na solução proposta neste trabalho foi adotado o padrão SCORM, embora seja possível expandir a solução e, de maneira automatizada, utilizar diferentes padrões para diferentes LMS.

O SCORM (*Sharable Content Object Reference Model*) é um dos principais padrões internacionais usados para a troca de objetos de aprendizagem e praticamente todos os LMS o implementam. O SCORM tem como objetivo prover meios técnicos para que os objetos de conteúdo possam ser facilmente compartilhados entre os diversos ambientes de aprendizagem, ou seja, além de serem reutilizáveis, os objetos de aprendizagem criados também devem ser interoperáveis.

O padrão SCORM descreve as seguintes especificações[4]:

- Como criar um conteúdo de aprendizagem baseado em Web que pode ser disponibilizado e utilizado em diferentes Sistemas LMS compatíveis.
- O que um LMS precisa fazer para disponibilizar e utilizar conteúdos no padrão SCORM.

4- Proposta Senac

O Senac São Paulo vem desenvolvendo, com qualidade, cursos corporativos a distância com o apoio da ferramenta Adobe Flash [5], que possibilita experiências audiovisuais através dos navegadores de internet (browsers).

No entanto, as demandas do mercado e o ritmo acelerado de trabalho exigem que os cursos sejam produzidos em cada vez menos tempo. Além disso, nem sempre a equipe de produção responsável pelo desenvolvimento dos cursos está disponível para novas demandas de trabalho. Daí a importância não só de reutilizar conteúdos já produzidos anteriormente como também desenvolver uma metodologia de trabalho que simplifique o processo de produção dos conteúdos, diminuindo a dependência da equipe de produção.

Com o passar do tempo, constatou-se que não havia uma padronização no formato dos cursos produzidos, e, embora pedagogicamente os conteúdos fossem pensados para serem reutilizáveis, essa possibilidade demandaria um grande retrabalho para adaptar visualmente e estruturalmente os materiais, uma vez que navegação, cores e formato dos mesmos não “conversavam” entre si.

Pensando em todas essas questões, foi feita uma parceria das equipes pedagógica, de produção e de tecnologia para pensar uma proposta de padronização de alguns elementos imutáveis dentre os cursos corporativos. Com o tempo essa idéia evoluiu para um padrão ainda mais organizado e autônomo de desenvolvimento dos cursos.

O projeto consiste em uma espécie de gerenciador de conteúdos, aqui chamado de *container*, cuja função é carregar objetos (neste caso, arquivos no formato swf), gerenciar a relação existente entre cada um desses objetos – ou seja, como será a navegação entre esses conteúdos de acordo com o que foi definido pelo web designer – e prover o armazenamento das configurações do usuário e de sua vivência ao acessar esses objetos, abrangendo suporte ao padrão SCORM. Uma das vantagens desse modelo é permitir que o web

designer tenha maior liberdade para criar as interfaces gráficas, que podem ser amplamente customizadas sem perder suas funcionalidades.



Figura 1. Componentes do *container*

A Figura 1 mostra um exemplo de curso utilizando o *container*, indicando os seguintes componentes:

1. Botão de pausa/continuação, utilizado para interromper e dar continuidade ao conteúdo apresentado, a qualquer momento.
2. Setas de navegação para que o usuário possa avançar ou voltar entre as telas de conteúdos apresentadas.
3. Barra de navegação interna para que o usuário possa, em uma mesma tela, adiantar ou retroceder na locução/animação proposta. Para tanto, pode utilizar as setas do teclado ou arrastar o marcador com o mouse.
4. Barra de progresso para que o usuário se localize dentro do curso/módulo. Esta barra indica a quantidade total de telas, bem como a posição da tela atual. Ao clicar em voltar ou avançar, uma tela nova ou já visitada é carregada e devidamente indicada nessa barra de progresso.
5. Nome do curso.
6. Nome do módulo/aula.

Embora o usuário navegue entre as telas por meio dos componentes apresentados, o gerenciamento dessa navegação é todo feito pelo *container*, através de um arquivo de configuração, evitando que o web designer entre em contato com a programação responsável por esse gerenciamento.

Neste modelo, por meio da configuração do *container*, é possível definir as seguintes restrições de navegação no conteúdo:

- **Obrigatoriedade apenas na primeira interação** - o usuário não poderá avançar através das setas de navegação ou mesmo utilizar a barra de navegação interna até que ele tenha

percorrido aquele conteúdo. Em outras palavras, o usuário só tem liberdade de navegação em conteúdos já vistos.

- **Sempre obrigatório** - o usuário só tem liberdade de navegação na tela na qual ele se encontra e caso já tenha percorrido todo o conteúdo da mesma.
- **Livre** - o usuário pode avançar ou regredir à vontade.

Outra capacidade do *container*, poupando mais uma vez que as equipes de produção e programação se unam, é determinar como as informações serão salvas (incluindo a implementação do SCORM) permitindo que o usuário retorne em um determinado ponto do conteúdo.

As opções de armazenamento do *container* se restringem a:

- Salvar o estado **a cada tela** concluída pelo usuário.
- Salvar o estado **a cada módulo** concluído pelo usuário.
- **Não salvar.**

Caso o *container* não consiga encontrar o SCORM, as configurações são salvas na máquina do usuário através do próprio Adobe Flash.

A estrutura do *container* é composta de quatro componentes: duas classes, Container.as e PersistenceManager.as (arquivos .as referem-se a ActionScript) que são responsáveis pela lógica e armazenamento do *container*; um arquivo de configuração, course.xml, onde são definidas restrições de navegação, armazenamento e os arquivos a serem carregados; dois arquivos JavaScript, SCOFuctions e APIWrapper, próprios para comunicação com o SCORM / LMS; e um arquivo principal, o container fla.

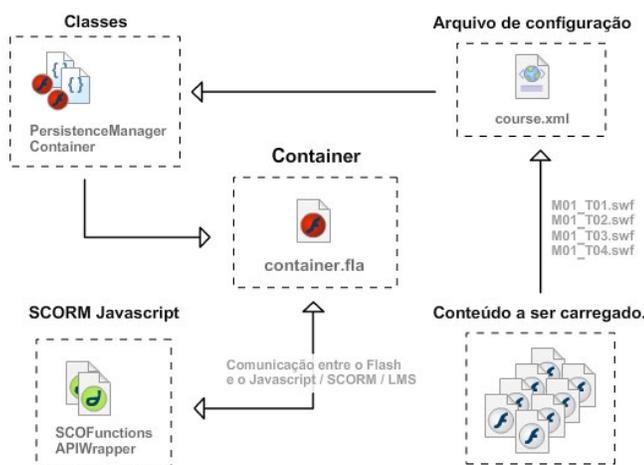


Figura 2. Estrutura detalhada do *container*

Como mostra a Figura 2, o container fla importa a classe Container que executa os métodos necessários para carregar o conteúdo. Ao fazê-lo, a classe Container carrega o arquivo de configuração (course.xml) que possui as referências para os conteúdos a serem carregados.

Ao carregar o primeiro conteúdo, o arquivo container.fla os apresenta para o usuário possibilitando ao mesmo navegar e, de acordo com a política definida no arquivo de configuração, fazer com que ocorra uma comunicação entre o Javascript/SCORM/LMS, gravando assim informações sobre a sessão do usuário.

Para permitir a reusabilidade, o arquivo container.fla possui todos os elementos gráficos e a chamada para a classe Container, que dá início ao funcionamento do container.

Na biblioteca do arquivo container.fla todos os elementos gráficos são separados por pastas.

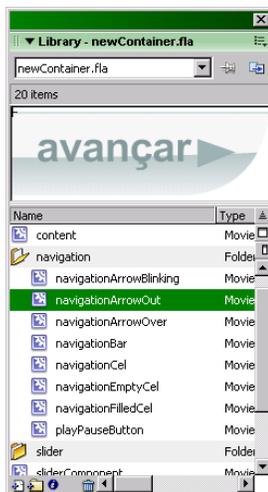


Figura 3. Estrutura de *MovieClips* no Flash

Como mostra a Figura 3, o web designer pode simplesmente abrir o *MovieClip* que representa determinado elemento da interface e modificá-lo como bem entender. Mesmo a disposição dos elementos na tela pode ser modificada, ou seja, o designer tem liberdade para posicionar a barra de navegação na parte superior da tela, por exemplo, sem que isso afete as funcionalidades da mesma. Observe que isso é válido para qualquer componente da interface.

A Figura 4 apresenta o mesmo conteúdo com interfaces completamente diferentes, porém utilizando os mesmos componentes.



Figura 4. Mesmo conteúdo, interfaces diferentes

Como já foi mencionado, há um arquivo de configuração no formato XML que define as restrições de navegação, armazenamento e os arquivos a serem carregados. Devido a algumas restrições do próprio Adobe Flash, foi necessário adotar uma solução alternativa para a definição dos arquivos a serem carregados. O *container* utiliza um preceito de padrão de nomes para encontrar os arquivos.

Para facilitar o entendimento segue uma linha do arquivo de configuração:

```
<module name="first module" pattern="M01_T\n.swf" length="4" />
```

Essa linha do arquivo de configuração define que o nome do módulo é “first module”, o padrão de nomeação é “M01_T\n.swf”. Observe que o caractere “\n” será substituído pelo container por números (01, 02, 03...n) dependendo do tamanho definido no atributo length, que, nesse caso, é igual a 4. Portanto, no exemplo acima os arquivos: M01_T01.swf, M01_T02.swf, M01_T03.swf e M01_T04.swf serão carregados pelo container.

Como se pode notar, a linha do arquivo de configuração apresentada está envolvida por uma tag chamada module, que pode ser repetida inúmeras vezes. Cada módulo possui sua própria barra de navegação e, ao terminar um módulo, uma nova barra de navegação é carregada.

Para fazer uso da restrição de navegação e/ou armazenamento é necessário utilizar uma outra tag chamada config, como mostra o exemplo a seguir:

```
<config navigationMode="free" lessonLocationMode="screen"/>
```

A tag config possui os atributos navigationMode e lessonLocationMode.

O atributo navigationMode se refere à forma de navegação do usuário, onde os valores podem variar entre “free” – sem restrições, “first time” – restrita apenas na primeira visita e “ever” – restrita sempre.

O atributo lessonLocationMode possui esse nome devido ao modo como o SCORM armazena as informações da sessão e se refere à forma de armazenamento, que pode variar entre “screen” – salvar cada conteúdo carregado, “module” – salvar cada módulo carregado e "" (vazio ou omissão do atributo) – não salvar.

A última configuração possível é o título do curso feito através da tag title. Por exemplo:

```
<title>First Course</title>
```

Unindo todas essas tags é possível formar um xml adequado para o arquivo de configuração do container. Segue abaixo um exemplo simples:

```
<?xml version="1.0" encoding="UTF-8"?>
<course>
  <title>Exemplo</title>
  <config navigationMode="free" lessonLocationMode="screen"/>
  <modules>
```

```
<module name="mod1" pattern="M01_T\n.swf" length="4" />
<module name="mod2" pattern="M02_T\n.swf" length="6" />
    <module name="mod3" pattern="M02_T\n.swf" length="3" />
</modules>
</course>
```

5- Considerações finais

Para as equipes envolvidas com a produção dos objetos/cursos, o benefício mais aparente da utilização do novo modelo de desenvolvimento de objetos reutilizáveis está no ganho de produtividade, uma vez que, adotado esse padrão, torna-se desnecessário o envolvimento da equipe de tecnologia/programação para as configurações referentes ao padrão SCORM. Sendo assim, o tempo anteriormente gasto com a configuração dos cursos pode ser utilizado para aprimorar questões de conteúdo e recursos.

A equipe de produção pode modificar a estrutura de navegação e layout, de acordo com cada identidade visual de diferentes cursos, sem que precise programar ou duplicar códigos. Todos os controles estão em um único local, que irá gerenciar todo o conteúdo de acordo com as necessidades específicas de cada curso.

Além disso, há diversas vantagens nesta característica dos objetos de aprendizagem. As instituições de ensino podem aproveitar de uma forma muito mais otimizada a troca de conteúdos, evitando, por exemplo, que professores desenvolvam conteúdos em duplicidade ou muito parecidos, além de trazer a possibilidade de um padrão institucional para os conteúdos.

Empresas provedoras de conteúdo ganham em produtividade, montando um repositório de objetos que podem ser utilizados na construção de novos cursos.

Para empresas privadas, além da possibilidade de um padrão institucional, ganha-se mais liberdade na escolha de seus fornecedores, evitando que cada um de seus treinamentos tenham características completamente diferentes. Além disso, é possível o desenvolvimento de treinamentos com conteúdos realmente necessários para cada colaborador, otimizando o tempo e a qualidade dos treinamentos.

Essa é uma proposta inicial, que tem como objetivo otimizar e aprimorar a produção de objetos levando em consideração aspectos como interface e navegação. O objetivo agora é aprimorar e expandir as funcionalidades desta proposta para configurações mais automáticas e completas, podendo possibilitar, por que não, a mudança de textos (como alteração de idiomas), diferentes formas de distribuição, com diferentes abordagens pedagógicas, de acordo com o perfil ou escolha do usuário [1].

Buscamos agora iniciar a discussão e troca de idéias e experiências com a comunidade de EaD para aprimorar a proposta aqui apresentada e utilizar ao máximo os benefícios do uso de objetos de aprendizagem.

7- Referências bibliográficas

- [1] BARRITT, Chuck; ALDERMAN, F. Lee. *Creating a reusable learning objects strategy*. San Francisco: Pfeiffer, 2004.
- [2] IEEE Learning Technology Standards Committee. The Learning Object Metadata standard. Disponível em: <http://www.ieeeltsc.org/working-groups/wg12LOM/lomDescription/>. Acessado em 30 abr. 2007.
- [3] POLSANI, Pithamber R. (2003) Use and Abuse of Reusable Learning Objects. *Journal of Digital Information* V.3, n.4. Article No. 164, 2003-02-19 Disponível em: <http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani/>. Acesso em 30 abr. 2007.
- [4] ADL. "ADL Sharable Content Object Reference Model", Version 1.2, <http://www.adlnet.org/>.
- [5] FLASH. Disponível em www.adobe.com/products/flash. Acesso em 01 mai. 2007

Nome do arquivo: 552007115338PM.doc
Pasta: C:\ABED\Trabalhos_13CIED
Modelo: C:\Documents and Settings\Marcelo\Dados de aplicativos\Microsoft\Modelos\Normal.dot
Título: DESENVOLVENDO OBJETOS DE APRENDIZAGEM REUTILIZÁVEIS: UMA PROPOSTA TECNOLÓGICA PARA FACILITAR A ORGANIZAÇÃO E TROCA DE OBJETOS
Assunto:
Autor: silvia.fecorrea
Palavras-chave:
Comentários:
Data de criação: 5/5/2007 23:40:00
Número de alterações:4
Última gravação: 5/5/2007 23:47:00
Salvo por: Illusionland
Tempo total de edição: 9 Minutos
Última impressão: 24/8/2007 17:55:00
Como a última impressão
Número de páginas: 10
Número de palavras: 2.968 (aprox.)
Número de caracteres: 16.030 (aprox.)