

**Wellington de Oliveira Neto**

**Estudo e análise de um modelo de referência para conteúdos e ambientes de ensino e aprendizado a distância: SCORM**

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para obtenção do título de Bacharel

Orientador  
Prof. Rêmulo Maia Alves

Co-Orientador  
Prof. Anderson Bernardo  
dos Santos

Lavras  
Minas Gerais - Brasil  
2004



**Wellington de Oliveira Neto**

**Estudo e análise de um modelo de referência para conteúdos e ambientes de ensino e aprendizado a distância: SCORM**

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para obtenção do título de Bacharel

Avaliada em *17/01/2005*

---

Prof. Rêmulo Maia Alves  
(Orientador)

---

Prof. Anderson Bernardo dos Santos  
(Co-Orientador)

Lavras  
Minas Gerais - Brasil



## **Agradecimentos**

Agradeço aos meus pais, José Ferreira Neto e Daili Coelho de Oliveira Ferreira, por terem me ensinado a ser quem sou hoje. Aos meus irmãos Luanna e Herverton por todos os momentos que vivemos juntos, a minha namorada Yonitta, que sempre me apoiou e esteve do meu lado, não deixando que eu desanimasse, "Você é demais, TE AMO ". A toda minha família, minhas avós Anas, meu avô José João e meu avô Aristóteles.

Agradeço também aos meus amigos: Helvécio, Luis (O Bardo), Almesindo (Frangon), Iberê, Baltazar (Biltz), entre muitos outros de Dom Cavati. Aos companheiros de curso e amigos da SW, por tudo que vivemos juntos. Aos companheiros de 210, o melhor apartamento que existe: Alisson (Binha), Eduardo (Fiote), Pedro (Galo Cego), Paulinho, Tarcisio (Bera), Leo Shigueto (Shiga, Joãoponês, Oriental, entre outros apelidos), Gleimar (Gueimar), Tico e Teço (Esse dois do 209, os quais nós sustentávamos), e todos os outros moradores do "Brejão".

Agradeço a DEUS e por último mais não menos importante ao Jaspion, por salvar a terra.



*À memória de meu avô Aristóteles.*





## **Resumo**

Com o crescimento do E-learning no mundo e o surgimento de inúmeros softwares de gerenciamento de aprendizagem, grandes instituições e empresas americanas perceberam que isto poderia se tornar algo prejudicial, e passaram a estudar e a adotar padronizações, tanto para software LMS quanto para conteúdos próprios ou de terceiros. Este trabalho tem por objetivo fazer um estudo profundo de um padrão de referencias para criação de LMS, o padrão SCORM (Sharable Content Object Reference Model).



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Considerações iniciais . . . . .	1
1.2	Objetivos e justificativas . . . . .	1
<b>2</b>	<b>EAD - Ensino a Distância</b>	<b>3</b>
2.1	Conceito . . . . .	3
2.2	Histórico da EAD . . . . .	4
2.3	Vantagens e Limites da EAD . . . . .	5
2.4	E-Learning . . . . .	7
2.4.1	Introdução . . . . .	7
2.4.2	Evolução no Brasil . . . . .	8
2.4.3	Significados, aplicações e diferenças entre CMS, LMS e LCMS . . . . .	9
2.5	LMS's que suportam SCORM . . . . .	12
2.5.1	Moodle . . . . .	12
2.5.2	ATutor . . . . .	13
2.6	WebCT . . . . .	13
2.7	WebAula . . . . .	14
2.8	Outros Padrões Existentes . . . . .	14
2.8.1	AICC . . . . .	14
2.8.2	IMS . . . . .	15
2.8.3	IEEE - LTSC . . . . .	15
<b>3</b>	<b>Padrão SCORM</b>	<b>17</b>
3.1	Visão Geral . . . . .	17
3.2	Content Aggregation Model . . . . .	19
3.2.1	Content Model . . . . .	19

3.2.2	Meta-data . . . . .	21
3.2.3	Content Packaging . . . . .	22
3.3	Run-Time Environment . . . . .	26
3.3.1	Launch . . . . .	27
3.3.2	API . . . . .	28
3.3.3	Data Model . . . . .	33
3.4	Sequencing and Navigation . . . . .	43
3.4.1	Activity Tree . . . . .	43
3.4.2	Cluster . . . . .	45
3.5	Tracking Model . . . . .	46
<b>4</b>	<b>Conclusão</b>	<b>49</b>
4.1	Considerações Finais . . . . .	49
4.2	Sugestões para Trabalhos Futuros . . . . .	49
<b>6</b>	<b>Referências bibliográficas</b>	<b>51</b>

# Lista de Tabelas

2.1	Evolução histórica da EAD . . . . .	5
3.1	Código dos erros mais frequentes . . . . .	31
3.2	Descrição de alguns elementos de cmi.core . . . . .	36
3.3	Descrição de alguns elementos de cmi.objectives . . . . .	39
3.4	Descrição de alguns elementos de cmi.interactions . . . . .	40



# Lista de Figuras

3.1	SCORM como um conjunto de especificações. . . . .	18
3.2	Exemplo de Assets (extraída de [SCORM_CAM]). . . . .	20
3.3	Exemplo de SCO (extraída de [SCORM_CAM]). . . . .	21
3.4	Exemplo de arquivo PIF (extraída de [SCORM_CAM]). . . . .	23
3.5	Diagrama da seção Organizations (extraída de [SCORM_CAM]).	24
3.6	Exemplo de um imsmanifest.xml (extraída de [SCORM_CAM]). .	25
3.7	Run-Time Environment de uma plataforma de E-Learning (extraída de [SCORM_RTE]). . . . .	26
3.8	Diagrama de estados de um SCO visto pelo API Adapter (extraída de [SCORM_RTE]). . . . .	33
3.9	Exemplo de uma Activity Tree (extraída de [SCORM_SN]). . . .	44
3.10	Exemplo de Cluster (extraída de [SCORM_SN]). . . . .	45
3.11	Tracking Model (extraída de [SCORM_SN]). . . . .	47





# Lista de Abreviaturas

**ADL** Advanced Distributed Learning

**AICC** Aviation Industry CBT Committee

**CBT** Computer Based Training

**CMI** Computer Managed Instruction

**CMS** Content Management System

**EAD** Ensino a Distância

**IMS** Instructional Management Systems Project

**IEEE** Institute of Electrical and Electronics Engineers

**LCMS** Learning Content Management System

**LMS** Learning Management System

**SCORM** Sharable Content Object Reference Model



# Capítulo 1

## Introdução

### 1.1 Considerações iniciais

Com o crescimento mundial do e-learning e surgimento de inúmeras ofertas de softwares e cursos à distância mediados via internet, desenvolvidos a partir de diversas tecnologias distintas. Grandes instituições e empresas norte-americanas passaram a estudar e a adotar padronizações, tanto para software LMS quanto para conteúdos próprios ou de terceiros. Estes padrões permitem aos fabricantes reutilizar o material didático, rastrear resultados dos cursos e certificações e manter um histórico destes dados.

O SCORM é um conjunto unificado de padrões e especificações para conteúdo, tecnologias e serviços para e-learning. Define um modelo de agregação de conteúdo (content aggregation model) e um ambiente de execução (Run-Time environment) para objetos educacionais baseados na web.

### 1.2 Objetivos e justificativas

O objetivo deste trabalho é realizar um estudo sobre o modelo SCORM para criação de ferramentas de ensino a distância, um bom argumento para sua utilização no desenvolvimento de conteúdo para e-learning pode ser resumido no acrônimo "RAID", ou seja, reusabilidade, acessibilidade, interoperabilidade e durabilidade. Um dos objetivos do SCORM é propiciar a independência de plataforma na qual os objetos serão utilizados, assim como facilitar a migração de cursos entre dife-

rentes ambientes de gerenciamento de aprendizagem que sejam compatíveis com esse padrão. A migração de um curso "empacotado" utilizando as especificações do SCORM demanda esforço mínimo. Além disso, o conteúdo desenvolvido em conformidade com SCORM é independente de contexto, ou seja, funcionará em situações variadas, seja inserido em um ambiente de gerenciamento de aprendizagem ou como parte de um curso on-line publicado diretamente na web ou ainda em cenário híbrido.

O ensino a distância é a metodologia do momento. Não por questão de modismo mas sim de evolução. A sociedade moderna, em permanente mudança, exige estudo constante de todos nós. A educação convencional não consegue mais atender à demanda de formação e atualização profissional no atual sistema. Felizmente, a evolução tecnológica possibilita novas soluções na área da educação. Uma dessas soluções é o ensino a distância: caracterizado pela separação física entre o professor e o aluno mas que são conectados por uma tecnologia para comunicação bidirecional.

A padronização dos softwares de ensino a distância é de suma importância na concretização desta evolução e o SCORM (Sharable Content Object Reference Model), é um padrão recente, mas que está sendo bastante incorporado pelo mercado e instituições de ensino.

## Capítulo 2

# EAD - Ensino a Distância

Este capítulo estará apresentando o conceito de EAD, um breve histórico e algumas de suas vantagens e limites.

### 2.1 Conceito

Para Aretio, [ARETIO1994], educação à distância (EAD) é um sistema tecnológico de comunicação bidirecional, que substitui o contato pessoal professor/aluno, como meio preferencial de ensino, pela ação sistemática e conjunta de diversos recursos didáticos e pelo apoio de uma organização e tutoria, que possibilitam a aprendizagem independente e flexível dos alunos.

Keegan, [KEEGAN1991], em seu estudo sobre Educação à Distância sumariza os elementos que considera centrais para sua caracterização:

- pela separação do professor e aluno no espaço e/ou tempo.
- controle do aprendizado realizado mais intensamente pelo aluno do que pelo instrutor distante.
- comunicação entre alunos e professores é mediada por documentos impressos ou alguma forma de tecnologia.

Abraçando a tese usada por Preti em sua dissertação de que a EAD deve ser compreendida como uma "prática educativa situada e mediatizada, uma modalidade de se fazer educação, de se democratizar o conhecimento"[PRETI1996], onde o conhecimento deve estar disponível a quem se dispuser a conhecê-lo, independente do lugar, do tempo e de engessadas estruturas formais de ensino. Sem dúvida é uma alternativa pedagógica que hoje dispõe o educador e as instituições escolares.

## **2.2 Histórico da EAD**

Nunes, [NUNES1992], nos coloca como marco inicial da educação a distância as cartas escritas no início da era cristã para disseminar a boa nova do senhor, evoluindo para a invenção da imprensa por Guttenberg, talvez o grande marco inicial para a disseminação da palavra escrita, já que antes, todos os livros eram copiados manualmente demandando tempo e dinheiro.

Segundo o que nos é colocado por Nunes, [NUNES1992], as mensagens escritas portanto, constituíram-se na primeira estratégia de estabelecer comunicação personalizada quando a distância não permitia o encontro dos interlocutores. Do início do século XX até a Segunda guerra mundial, várias experiências foram adotadas, ocasião em que as metodologias aplicadas ao ensino por correspondência se desenvolveram melhor, e que posteriormente, foram fortemente influenciadas pela introdução de novos meios de comunicação de massa, como o rádio, que deu origem a projetos muito importantes, principalmente no meio rural.

O grande salto se dá a partir de meados dos anos 60, com a institucionalização de várias ações no campo da educação secundária e superior, começando pela Europa (França e Inglaterra) e se expandindo pelos demais continentes. Como grande marco da educação a distância podemos citar a criação da Open University da Inglaterra em 1969 sendo um modelo de sucesso até os dias de hoje.

Moore, [MOORE1996], nos apresentam a evolução histórica da EAD dividindo-a em 3 gerações que são colocadas na tabela. 2.1.

Tabela 2.1: Evolução histórica da EAD

Geração	Descrição
até 1970	Estudo por correspondência, no qual o principal meio de comunicação eram materiais impressos, geralmente um guia de estudo, com tarefas ou outros exercícios enviados pelo correio.
1970	Surgem as primeiras universidades abertas, com design e implementação sistematizadas de cursos a distância, utilizando além do material impresso, transmissões por televisão aberta, rádio e fitas de áudio e vídeo, com interação por telefone, satélite e TV a cabo.
1990	Esta geração é baseada em redes de conferência por computador e estações de trabalho multimídia.

A evolução da educação a distância neste início de século XXI está atrelado ao desenvolvimento dos meios de comunicação, principalmente da Internet; as novas redes de comunicação podem prover a um curto prazo acessos mais rápidos e confiáveis, permitindo um nível maior de interatividade. Neste aspecto, considerando que é a partir da ampliação dos níveis de interatividade que ampliaremos a qualidade educacional desta modalidade, tem-se nesta tecnologia, excelentes perspectivas para o desenvolvimento deste modelo.

### 2.3 Vantagens e Limites da EAD

Segundo Aretio, [ARETIO1994], a Educação a Distância proporciona as seguintes vantagens:

- **Abertura:** eliminação ou redução das barreiras de acesso aos cursos ou nível de estudos; diversificação e ampliação da oferta de cursos; oportunidade de formação adaptada às exigências atuais, às pessoas que não puderam

freqüentar a escola tradicional.

- **Flexibilidade:** ausência de rigidez quanto aos requisitos de espaço (onde estudar?), assistência, às aulas e tempo (quando estudar?) e ritmo (em que velocidade aprender?); eficaz combinação de estudo e trabalho; permanência do aluno em seu ambiente profissional, cultural e familiar; formação fora do contexto da sala de aula.
- **Formação permanente de pessoal:** atendimento às demandas e às aspirações dos diversos grupos, por intermédio de atividades formativas ou não; desenvolvimento da iniciativa, de atitudes, interesses, valores e hábitos educativos; capacitação para o trabalho e superação do nível cultural de cada aluno.
- **Eficácia:** aluno, centro do processo de aprendizagem e sujeito ativo de sua formação, vê respeitado o seu ritmo de aprender; formação teórico-prática, relacionada à experiência do aluno, em contato imediato com a atividade profissional, que se deseja melhorar; conteúdos instrucionais elaborados por especialistas e a utilização de recursos multimídia; comunicação bidirecional freqüente, garantindo uma aprendizagem dinâmica e inovadora.
- **Economia:** redução de custos em relação aos dos sistemas presenciais de ensino, ao eliminar pequenos grupos, ao evitar gastos de locomoção de alunos, ao evitar o abandono do local de trabalho para o tempo extra de formação, ao permitir a economia em escala; a economia em escala supera os altos custos iniciais.

Ainda o próprio Aretio, [ARETIO1994], cita algumas limitações:

- limitação em alcançar o objetivo da socialização, pelas escassas ocasiões para interação dos alunos com o docente e entre si.
- empobrecimento da troca direta de experiências proporcionada pela relação educativa pessoal entre professor e aluno.



- perigo da homogeneidade dos materiais instrucionais todos aprendem o mesmo, por um só pacote instrucional, conjugado a poucas ocasiões de diálogo aluno/docente, pode ser evitado e superado com a elaboração de materiais que proporcionem a espontaneidade, a criatividade e a expressão das idéias do aluno.

## **2.4 E-Learning**

Nesta seção será apresentada uma introdução ao E-Learning, bem como a apresentação de alguns conceitos de soluções ligadas ao mesmo.

### **2.4.1 Introdução**

Os micros são parte integrante de nossas vidas, ou melhor dizendo, estão se tornando essenciais para o nosso aprendizado. A maioria das pessoas acredita que os treinamentos baseados no computador são inovações educacionais recentes. Entretanto, existem há cerca de 30 anos.

Hoje, a tecnologia é mais sofisticada e a Internet alterou, fundamentalmente, as paisagens econômicas de maneira tão radical que agora é possível dar saltos significativos na utilização da tecnologia para o aprendizado.

Há poucos anos apenas algumas pessoas tinham ouvido falar no termo e-learning. Mas, aconteceu uma pequena revolução. Se antes falava-se em ensino com tecnologia, a palavra hoje é treinamento on-line.

Aos poucos, a introdução do e-learning dentro de nossas casas, das empresas e das próprias escolas e universidades adicionou um novo significado para o treinamento e as possibilidades para entrega e formação de conhecimento.

Os cursos e treinamentos são disponibilizados num compasso acelerado na rede e os estudantes têm acesso a um novo mundo para a transferência de conhecimento. A discussão foi ampliada e hoje a questão não é mais se as empresas implementarão o aprendizado on-line, mas se elas o farão da maneira correta. Ter a tecnologia certa e fornecer bons programas de aprendizado utilizando essa tecnologia é essencial, mas não o suficiente.

Entende-se por E-Learning a utilização das Tecnologias de Informação e Comunicação (TIC) como meio de transmitir conteúdos e promover a aprendizagem.

Os cursos e programas de formação servidos por uma plataforma de e-Learning podem fazer uso de qualquer formato eletrônico para expor os seus conteúdos

e usar diversos meios para divulgá-los: CD-ROM, DVD, Internet, Intranet, etc. Distinguem-se habitualmente duas modalidades de formação:

- A modalidade síncrona, ou seja, um componente de formação em tempo real, que promove a interação através de voz, imagem e dados, entre alunos e docentes numa "sala de aula virtual", independente do local onde se encontrem.
- A modalidade assíncrona, ou seja, sem a possibilidade de interação em tempo real e que proporciona o acesso aos conteúdos de forma individualizada. Nestes casos, se existe interação com os formandos ou formadores, ela é realizada em diferido através de e-mails ou fóruns de discussão.

Algumas das vantagens do E-Learning são:

- Eliminam as barreiras da distância e do acesso aos professores/formadores.
- Conferem a cada indivíduo ou organização o privilégio de gerir de forma personalizada os seus tempos de formação.
- Otimizam o investimento na formação através da redução de custos associados à deslocação e à ausência do local de trabalho dos seus colaboradores.
- Permitem a participação nas sessões de formação através de horários mais flexíveis e ajustados às necessidades de cada utilizador (24 horas por dia).

Podemos citar como desvantagem, além da falta de comprometimento, o fato de se ignorar que todos os cursos requerem um esforço pessoal, não suprido pela tecnologia. Em alguns contextos, é possível adquirir conhecimento através de auto-aprendizado à distância, mas, na maioria das vezes, faz-se necessário interagir com outras pessoas. Mais cedo ou mais tarde, o que se aprende precisa ser contextualizado com outras pessoas.

#### **2.4.2 Evolução no Brasil**

Em 2002 e 2003 ninguém mais se perguntava se o E-Learning iria substituir as formas tradicionais de ensino, ou seja, as salas de aula. Todos já entendiam perfeitamente que o E-Learning veio para complementar e ampliar os momentos de aprendizado, assim como o e-mail o fez para as comunicações tradicionais em papel. Mas e daqui para frente, como será a evolução do E-Learning?

Dependerá do contexto em que as organizações estiverem dos objetivos que deveremos atingir nos próximos anos e da tecnologia que pudermos utilizar. Dependerá também do quanto estivermos dispostos a investir para termos profissionais melhor qualificados para fazer frente ao constante desafio de mercado com os quais nos defrontamos diariamente.

O mercado de E-Learning cresce a passos largos em todo mundo e no Brasil. Prova dessa demanda é o aumento no número de organizações que usam E-Learning como ferramenta de Tecnologia e Desenvolvimento. De um lado, temos as empresas que se empenham em criar um diferencial em meio à concorrência, procurando qualificação profissional. Do outro, especialistas em educação corporativa e universidades incluem em seu catálogo de serviços o treinamento a distância, via Intranet ou Internet. Empresas de tecnologia correm atrás desse filão. Com isso a educação a distância se consolida cada vez mais no país.

Os empresários querem garantir a competitividade no mercado e, para tanto, precisam investir na atualização, treinamento e aperfeiçoamento constante de seus funcionários, pré-condições para o desenvolvimento e a retenção de talentos. A atual visão do empresariado gerou nos Estados Unidos uma enorme procura pelo ensino à distância que, desde 1997, vem possibilitando a capacitação com eficácia e baixo custo. No Brasil, só recentemente, empresas dos mais diversos setores, preocupadas em promover o aprimoramento e motivação dos seus empregados, estão descobrindo o treinamento pela Internet.

### **2.4.3 Significados, aplicações e diferenças entre CMS, LMS e LCMS**

É muito comum encontrar informações na Internet sobre soluções classificadas como CMS (Content Management Systems), LMS (Learning Management Systems), LCMS (Learning Content Management Systems), e outras. Mas afinal, o que diferencia cada uma das categorias acima?

Além de compreender as diferenças técnicas e as aplicações específicas de cada tipo de solução, é fundamental entender e prever de que maneira cada uma afeta professores, instrutores, estudantes e treinandos.

#### **LMS - Learning Management System**

Um LMS ou Sistema Gerenciador do Processo de Aprendizagem tem o objetivo principal de simplificar a administração dos programas de treinamento e educação em uma organização. O sistema auxilia funcionários ou estudantes a planejarem

seus processos de aprendizagem individualmente, e ainda permite que os mesmo colaborem entre si através da troca de informações e conhecimentos.

No caso dos administradores, o sistema auxilia a análise, a disponibilização das informações, o rastreamento de dados, e a geração de relatórios sobre o progresso dos participantes. A maioria dos sistemas tipo LMS não possuem recursos que permitem a rápida e simples criação de conteúdos instrucionais, e este é um dos principais motivos pelo qual a maioria das empresas fornecedoras tem procurado oferecer ferramentas complementares ou trabalhar com parceiros de conteúdos.

### **CMS - Content Management System**

Um CMS (Sistema Gerenciador de Conteúdos) é um termo normalmente utilizado pela mídia eletrônica. Seu objetivo é simplificar e agilizar os processos de criação, publicação e administração de conteúdos (artigos, relatórios, imagens, mídias, cursos, etc.).

Podemos considerar como exemplo, um site de notícias fictício que possua 100 repórteres espalhados pelo mundo. Com o objetivo de evitar um verdadeiro pesadelo durante as publicações em virtude da dificuldade de consolidar diversos artigos provenientes de vários países em diferentes formatos e horários, este site pode utilizar um CMS para gerenciar as informações e padronizar processos.

Este gerenciamento pode ainda incluir de modo mais detalhado:

- Separação entre conteúdos e apresentação: os repórteres precisarão se concentrar somente em preparar os conteúdos (através de uma ferramenta como o MS Word por exemplo), e não terão de se preocupar com o visual das informações. Isto se torna possível e simples através da criação de templates com imagens e formatação próprias nas quais o próprio repórter pode incluir as informações de modo simples.
- Processos de trabalho dinâmicos: os artigos enviados pelos repórteres são primeiramente aprovados pelos editores e depois são publicados. E uma vez publicados, estes artigos permanecem acessíveis para o público por um determinado período, e depois são automaticamente arquivados pelo sistema.

Percebe-se portanto, que os conteúdos são criados através de componentes ou partes individuais que podem ser trabalhados de modo mais flexível. No caso do

site exemplificado, estas características permitem uma experiência mais interessante para o leitor, pois é mais fácil estruturar os conteúdos de modo mais específico para os desejos ou necessidades de cada cliente. Um profissional interessado principalmente em dados e informações do mercado financeiro poderá, por exemplo, receber informações mais detalhadas sobre este tema e menos notícias sobre esportes.

O mesmo princípio pode ser aplicado para os cursos que venham a ser oferecidos através do e-Learning em uma organização. Cada componente de informação ou objeto de aprendizagem (Learning Object) pode ser disponibilizado para o público de modo mais segmentado e focado. Desta forma é mais fácil atingir a pessoa certa, na hora certa com o conteúdo certo, que é um dos princípios fundamentais do e-Learning.

### **LCMS - Learning Content Management System**

Um LCMS ou Sistema Gerenciador de Conteúdos e Aprendizagem é uma solução, na maioria das vezes totalmente web, que é utilizada para criar, aprovar, publicar e gerenciar conteúdos instrucionais (geralmente chamados de objetos de aprendizagem).

Um LCMS combina os recursos de administração e gerenciamento de um tradicional LMS com as funcionalidades de criação e customização de conteúdos e cursos de um CMS. Nele é possível encontrar bibliotecas repletas de objetos de aprendizagem que podem ser utilizados independentemente ou em conjunto como parte de cursos instrucionais mais completos.

Assim como no caso de sistemas do tipo CMS, a adoção de um sistema LCMS também envolve o estabelecimento de processos de trabalho. Abaixo estão alguns exemplos de atividades:

- Os mediadores instrucionais (Instructional Designers) geralmente criam os objetos e os cursos com objetivos específicos ou estruturam novos cursos com objetos já existentes.
- Os editores, que podem ser mediadores seniores ou gerentes de projeto são os que avaliam os objetos e os cursos e definem a aprovação ou rejeição dos mesmos. Se aprovados os conteúdos são publicados, e se rejeitados retornam para uma revisão por parte da equipe de desenvolvimento.
- Regras de personalização de cursos devem ser definidas de acordo com os

perfis pré-definidos, com os objetivos de indivíduos e grupos, e com os objetos de aprendizagem disponíveis.

- Conteúdos que não mais se apliquem por estarem ultrapassados ou por serem de difícil reformulação para atualização devem ser arquivados ou apagados da biblioteca de conteúdos do sistema.

## **2.5 LMS's que suportam SCORM**

Como já foi dito acima o padrão SCORM vem crescendo em uso e já são vários os LMS's que o suportam (hoje em dia são poucos os LMS's que não o suportam), aqui serão citados alguns:

### **2.5.1 Moodle**

O Moodle é um software para produzir e gerenciar atividades educacionais baseadas na Internet e/ou em redes locais. É um projeto de desenvolvimento contínuo projetado para apoiar o social-construtivismo educacional. Conjuga um sistema de administração de atividades educacionais com um pacote de software desenhado para ajudar os educadores a obter alto padrão de qualidade em atividades educacionais on-line que desenvolvem.

A palavra Moodle era originalmente um acróstico para "Modular Object-Oriented Dynamic Learning Environment" que é principalmente útil aos pesquisadores e acadêmicos de educação.

É um software de Open Source, o que significa é livre para carregar, usar, modificar e até mesmo distribuir (sob a condição do GNU)[GNU]. o Moodle funciona sem modificação em Unix, Linux, Windows, Mac OS X, Netware e em qualquer outro sistema que suporte a linguagem PHP, podendo portanto ser incluído na maioria dos provedores de hospedagem. Os dados são armazenados em um único banco de dados: Funcionam mais eficientemente com MySQL e PostgreSQL, mas também pode ser usado com Oracle, Access, Interbase, ODBC e outros.

O suporte do padrão SCORM no Moodle é feito por meio de um módulo chamado SCORM, que ainda está em fase de desenvolvimento.

### **2.5.2 ATutor**

O ATutor é um sistema de gerência de conteúdo e aprendizagem (LCMS), Open Source, projetado visando acessibilidade e adaptabilidade. Os administradores podem instalar ou atualizar o ATutor em poucos minutos, também desenvolver temas e personalizar o ambiente ATutor. Os educadores podem rapidamente montar, empacotar, e redistribuir o material didático. Os estudantes aprendem em um ambiente adaptável da aprendizagem.[ATutor]

Adotou as especificações do IMS/SCORM Content Packaging, permitindo aos desenvolvedores criar conteúdo reutilizável que pode ser trocado entre sistemas de aprendizagem diferentes. Podendo ser importado um conteúdo criado em outro sistemas em conformidade com IMS ou SCORM para o ATutor, ou vice versa.

A versão atual do ATutor suporta apenas SCORM Content Packaging, não suportando o SCORM Run-Time Environment.

## **2.6 WebCT**

O WebCT foi desenvolvido no departamento de Ciência da Computação da University of British Columbia, em projeto chefiado por Murray W. Goldberg. É uma ferramenta que facilita a criação de sofisticados ambientes educacionais baseados em WWW. Ele permite a educadores fazer o design da aparência das páginas de curso, providencia um conjunto de ferramentas educacionais para o aluno que podem ser facilmente incorporadas em qualquer curso. Além disso, fornece um conjunto de ferramentas que assiste o professor (ou instrutor) na tarefa de administração do curso.

Toda interação com WebCT se dá através do browser. Isso inclui a administração do servidor, criação do curso, acesso do estudante e acesso do professor. A vantagem da ferramenta é incluir uma interface simples que pode ser usada de qualquer lugar através de um computador ligado na rede. Enquanto isso é importante para as quatro classes de usuários do WebCT, é especialmente importante para os estudantes, pois o acesso aos recursos de aprendizado torna-se independente do tempo e da localização.

## **2.7 WebAula**

O LMS WebAula é um produto formado por soluções integradas de gerenciamento de aprendizagem, conhecimento e conteúdos on-line, compatível com os padrões SCORM (versões 1.1 e 1.2) e AICC, sendo seu LMS capaz de importar cursos desenvolvidos em ambos os padrões (via Gestor de Conteúdos) e disponibilizar estes cursos com exercícios, provas e notas.

Com esse avanço, a plataforma E-Learning WebAula torna-se a primeira genuinamente brasileira a ser totalmente compatível com milhares de cursos para internet que já estão prontos no mundo inteiro.[WEBAULA]

## **2.8 Outros Padrões Existentes**

Nesta seção serão apresentados alguns outros padrões para criação de conteúdo para o ensino a distância. Existem mais padrões no mercado mais aqui serão abordados os padrões: AICC, IMS e IEEE. Uma vez que o SCORM faz uso dos mesmos em seu modelo.

### **2.8.1 AICC**

Os esforços de padronização mais antigos de nossa jovem indústria começaram na aviação, quando o e-Learning era mais bem conhecido por termos como Computer-Based Training (CBT) e Computer-Managed Instruction (CMI). O Aviation Industry CBT Committee (AICC), [AICC], estava estabelecido de uma maneira que montadoras como a Boeing, que compram partes de aviões de vários fabricantes diferentes, poderiam ter certeza de que cada um dos cursos de treinamento de seus fornecedores trabalhava da mesma maneira que os outros.

Os padrões AICC permitiram aos fabricantes reutilizar o material de treinamento, rastrear resultados de cursos e certificações, e transmitir novamente os dados para o LMS.

O padrão AICC trabalhou tão bem que suas regras foram no muito bem aceitas, e seu selo de aprovação é buscado por diversas empresas de e-Learning, muito além do mundo da aviação. Mas muitos dos padrões AICC foram desenvolvidos antes da Internet tornar-se o veículo chefe para a "entrega" de treinamento e aprendizagem on-line.

Outros órgãos então começaram a trabalhar para definir parâmetros de objetos de conteúdo baseados na Web, tais como o IMS Global Learning Consortium



(IMS) e o Institute of Electrical and Electronic Engineer's (IEEE) Comitês de Padrões de Tecnologia de Aprendizado.

### **2.8.2 IMS**

O IMS (Instructional Management Systems Project), [IMS], é um consórcio de instituições educacionais, companhias de software e editoras. O objetivo é promover a disseminação de especificações que permitam ambientes de aprendizado distribuído e união do conteúdo de diversos autores, ou seja interoperabilidade.

O IMS Content Packaging Information Model descreve as estruturas de dados responsáveis pela interoperabilidade de conteúdo baseado em Internet com ferramentas de criação de conteúdo e LMS (Learning Management Systems). O objetivo é definir um conjunto de estruturas que possam ser usadas na troca de conteúdo.

O arquivo Package Interchange File, pacote de troca, é um arquivo simples (.jar, .zip, .cab) que inclui o nível superior do manifesto, em um formato de entrega Web. O Package, pacote, representa a unidade utilizável do conteúdo, e tem instruções relevantes do exterior da organização do curso, podendo ser entregue independentemente, como um curso completo, ou coleção deles.

O Manifest, manifesto, é uma descrição em XML (Extensible Markup Language) dos recursos contidos nas instruções. Pode descrever parte do curso, ou o curso inteiro. Os recursos, Resource, descritos no manifesto são elementos físicos tais como páginas Web, arquivos textos, outras mídias, objetos de avaliação, etc.

Os elementos do manifesto podem ser obrigatórios (M 'Mandatory'), condicionais (C), ou Opcionais (O).

### **2.8.3 IEEE - LTSC**

Formado em 1996, o IEEE Learning Technology Standards Committee (LTSC), [IEEE] está padronizando várias áreas de tecnologia de ensino, não apenas acadêmico, mas também padrões para viabilizar comercialmente a educação à distância.

O padrão 1484.1 especifica o Modelo de Referência/Arquitetura para componentes baseados em sistemas CAI (Computer-Aided Instruction), incluindo as necessidades dos ILE - Intelligent Learning Environment e ITS (Intelligent Tutoring Systems).

O 1484.2 o modelo do aprendiz, sua sintaxe e semântica, caracterizando o aprendiz (estudante) e suas habilidades e conhecimentos.

O padrão 1484.3 contém o glossário. O padrão 1484.4 é o modelo de tarefa que caracteriza os aspectos estáticos de um projeto, incluindo descrição da tarefa, pré-requisitos, papéis, cronograma, recursos, objetivos de ensino, entregas e avaliações dos membros.

O 1484.5 é responsável pela Interface do Usuário. O padrão 1484.6 define o seqüenciamento do curso especificando linguagem e ambiente para gerenciamento de sessões entre os sistemas de educação à distância.

O padrão 1484.10 define o modo de intercâmbio entre dados de CBT (Computer-based Training). O padrão 1484.12 define os metadados dos objetos de aprendizado.

O padrão 1484.13 especifica a sintaxe, semântica, codificação, registro, resolução e autenticação do Identificador Único do Estudante.

O padrão 1484.17 especifica o empacotamento do conteúdo de aprendizado, tipicamente como uma coleção de componentes que pode ser copiada, transmitida, comprada, executada, e usada como uma unidade simples.

O padrão 1784.19 trata da Qualidade dos Sistemas visando o Aprendizado de longo-prazo.

O 1484.20 define as Competências entre os vários elementos para permitir criação, troca e reutilização de aplicações e componentes.

## Capítulo 3

# Padrão SCORM

### 3.1 Visão Geral

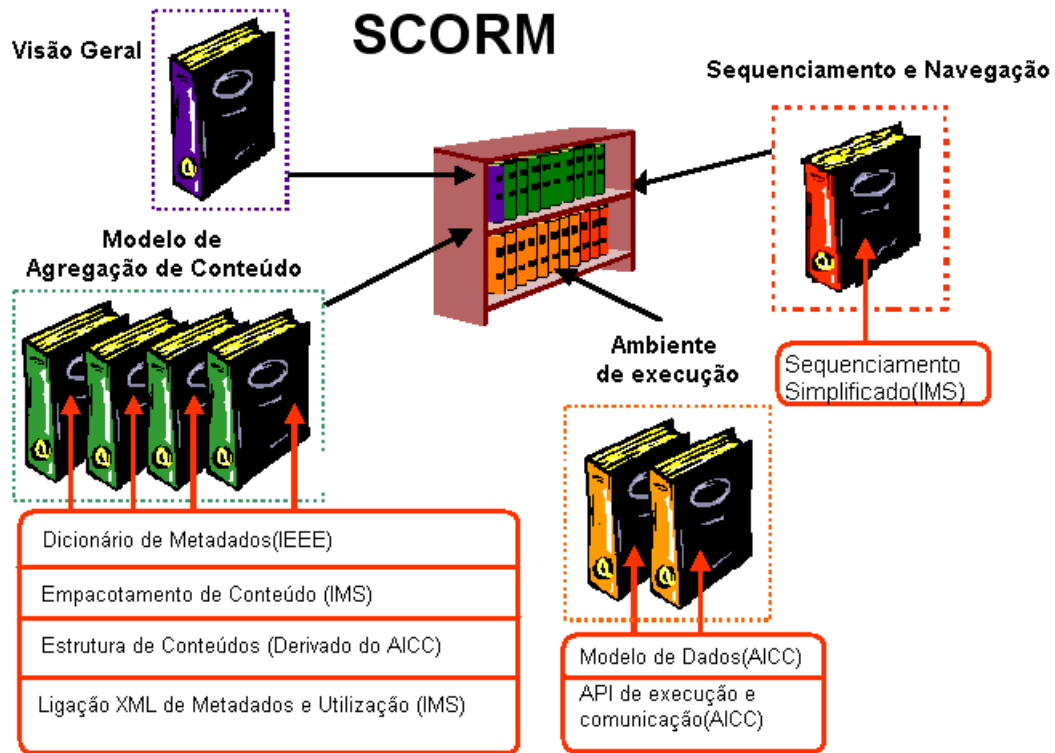
O padrão SCORM (Sharable Content Object Reference Model) pode ser definido de uma forma simples, como sendo um modelo que descreve um conjunto de especificações técnicas e de referências para apresentação de conteúdos de ensino via Web.

Criado pela ADL (Advanced Distributed Learning ) com o intuito de uniformizar as diversas implementações de sistemas de E-Learning que começaram a surgir, baseadas em diferentes tecnologias. O padrão SCORM aplica os desenvolvimentos atuais da tecnologia na construção de um modelo com o objetivo de produzir recomendações para a implementação consistente de sistemas de E-Learning.

Num sistema de E-Learning existem tipicamente os seguintes módulos e funcionalidades:

- Ferramentas de criação de recursos de ensino.
- Uma base de dados (repositório) para guardar os recursos de ensino.
- Um sistema para entregar os recursos de ensino aos alunos.
- Uma forma de acompanhar a progressão do aluno e de avaliar o seu desempenho e conhecimentos sobre o curso.

A figura a seguir ilustra o padrão SCORM como um conjunto de especificações:



**Figura 3.1:** SCORM como um conjunto de especificações.

O padrão SCORM distingue de forma muito clara as funções dos recursos de ensino das funções dos sistemas de gestão. Os recursos de ensino têm a designação de Sharable Content Objects (SCO) e o sistema de gestão é designado por Learning Management System (LMS).

Quando os formadores criam um curso trabalham apenas com os conteúdos de ensino (SCO). Posteriormente o LMS irá determinar a forma e a ordem pela qual o aluno irá ver os SCO's criados. Isto significa que os formadores têm que incluir nos SCO's instruções para indicar ao LMS quais os conteúdos este irá usar, a forma como os conteúdos estão organizados, a ordem pela qual devem ser apresentados

e qual a informação que desejam salvar relativamente a cada sessão.

O padrão SCORM divide-se em três partes:

- **Content Aggregation Model:** Define a forma como os conteúdos de ensino devem ser criados e agrupados.
- **Run-Time Environment:** Define a forma como o LMS disponibiliza os SCO's e como é que estes comunicam com o LMS, isto é, especifica um Modelo de Dados.
- **Sequencing and Navigation:** Define as sequências que o do curso e como os SCO's iram tratar os eventos de navegação.

## **3.2 Content Aggregation Model**

O Content Aggregation Model permite aos formadores agrupar os conteúdos que criarem seguindo um modelo que os LMS são capazes de interpretar.

O Content Aggregation Model é constituído pelos seguintes componentes:

- **Content Model:** Define os elementos onde podem ser guardados os conteúdos de ensino.
- **Meta-data:** Descreve o conteúdo dos recursos de ensino. Permite ao LMS organizar de forma conveniente os SCO's disponíveis e torna mais rápida e eficiente a pesquisa dos SCO's guardados.
- **Content Packaging:** Define como devem ser agrupados os conteúdos de ensino para permitir que sejam instalados em ambientes diferentes.

### **3.2.1 Content Model**

O Content Model define três componentes: Assets, Sharable Content Objects (SCO) e Content Aggregation.

#### **Assets**

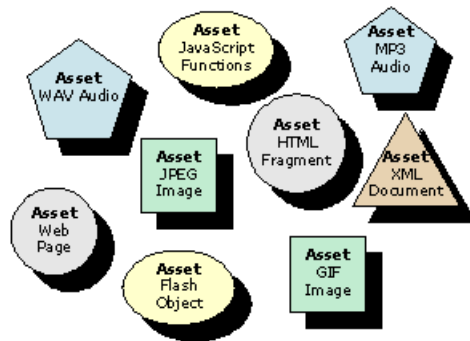
Os Assets constituem a forma mais básica de conteúdo de ensino. Podem fazer parte de um Asset: texto, imagens, som, páginas HTML e qualquer formato que possa ser entregue pela Web.

#### **Sharable Content Objects**

O Sharable Content Object (SCO) representa um conjunto de um ou mais Assets e inclui (obrigatoriamente) um mecanismo para comunicar com o LMS. Embora o LMS possa apresentar Assets, estes devem ser sempre inseridos em SCOs para que o LMS possa acompanhar a evolução dos alunos através dos dados passados pelo SCO. As comunicações entre os SCOs e o LMS serão analisadas na seção 3.3, quando tratarmos do Run-Time Environment.

Os SCOs devem ser pequenas unidades de ensino para potenciar a sua reutilização em diferentes contextos de ensino, ou seja, para permitir que o mesmo SCO possa ser utilizado em vários cursos. Quanto menor for o SCO mais facilmente poderá ser reintegrado noutra contexto.

A reutilização de conteúdos de ensino é um dos pontos-chave da filosofia do padrão SCORM. Com o SCORM pretende-se não só a possibilidade de transferir cursos entre LMS mas também facilitar a construção desses mesmos cursos através da utilização de unidades já existentes.



**Figura 3.2:** Exemplo de Assets (extraída de [SCORM\_CAM]).

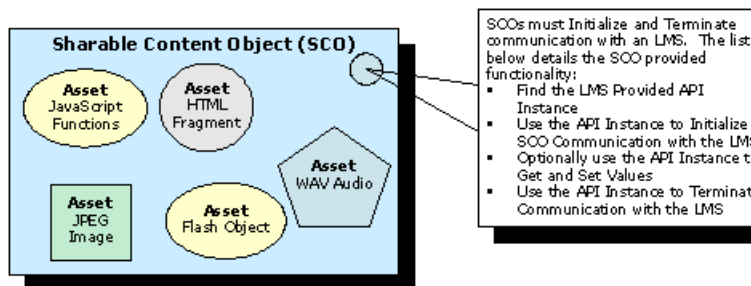


Figura 3.3: Exemplo de SCO (extraída de [SCORM\_CAM]).

### Content Aggregation Model

O Content Aggregation é um mapa usado para reunir os recursos de ensino numa unidade de instrução coesa (por exemplo, um curso, um capítulo, um módulo, etc.) e lhes dar uma estrutura bem definida. É esta estrutura que define a sequência pela qual os conteúdos de ensino serão apresentados aos alunos.

A sequência e a navegação entre os conteúdos de ensino são definidas na estrutura através da atribuição de pré-requisitos a cada conteúdo. Cabe ao LMS interpretar a sequência descrita e controlar a navegação da forma pretendida.

Para preservar a reutilização de conteúdos, a navegação entre SCO é sempre feita através do LMS. Em particular, um SCO não pode conter *links* para outros SCOs, porque nesse caso, se o SCO for reutilizado noutra curso, os SCOs para os quais contém *links* poderão não existir no novo curso. Os únicos *links* que o SCO pode ter são a nível interno, isto é, para páginas ou recursos existentes dentro do próprio SCO.

### 3.2.2 Meta-data

A meta-data consiste em informação acerca da informação. Concretamente no modelo SCORM, consiste em informação que descreve o conteúdo dos diversos componentes: Assets, SCO e Content Aggregation. A função da meta-data é disponibilizar um meio coerente de descrever o conteúdo de cada componente de modo a que este possa ser arquivado e pesquisado de uma forma rápida e eficiente.

A aplicação de meta-data no modelo SCORM respeita o padrão IEEE LTSC Learning Object Meta-Data (LOM)[IEEE] e utiliza a IMS Learning Resource Meta-data XML Binding Specification [IMS] para guardar a informação em formato XML.

### 3.2.3 Content Packaging

O Content Packaging consiste num conjunto de regras e normas para agregar conteúdos de ensino em blocos (packages), com o objetivo de permitir a transferência desses mesmos blocos entre sistemas diferentes: entre LMS's, entre as ferramentas de criação de conteúdos e o LMS, etc.

Os packages dividem-se em duas partes:

- Um documento especial XML que descreve os conteúdos e a organização do pacote (package).
- Os arquivos físicos respeitantes aos recursos de ensino.

Os pacotes (packages) criados são geralmente guardados num arquivo PIF (Package Interchange File) com o intuito de facilitar a sua distribuição pela Web. O PIF pode ter diversos formatos sendo os mais utilizados: zip, jar, tar e cab.

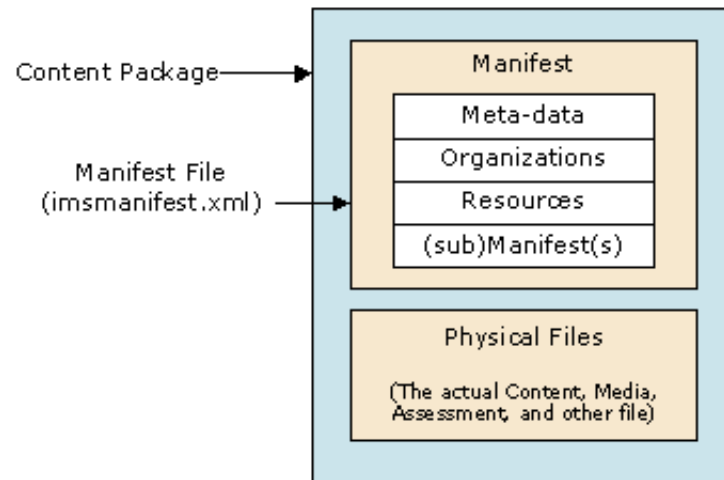
O documento XML constitui um manifesto dos diversos componentes do pacote, por isso mesmo, tem o nome *imsmanifest.xml*.

Do manifesto fazem parte os seguintes elementos:

- Meta-data sobre o bloco.
- Uma seção (opcional) com o nome Organizations que define a estrutura e o comportamento dos conteúdos de ensino.
- Uma lista de apontadores para os recursos guardados no bloco (Resources).

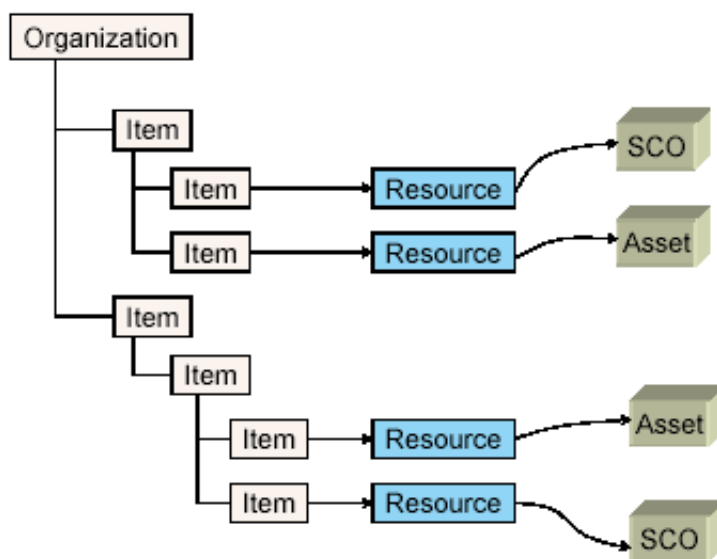


A figura seguinte ilustra a estrutura de um PIF:



**Figura 3.4:** Exemplo de arquivo PIF (extraída de [SCORM\_CAM]).

A seção Organizations permite organizar e estruturar os conteúdos de ensino como a figura seguinte ilustra:



**Figura 3.5:** Diagrama da seção Organizations (extraída de [SCORM\_CAM]).

Para concluir a seção apresenta-se na figura seguinte um Manifesto:

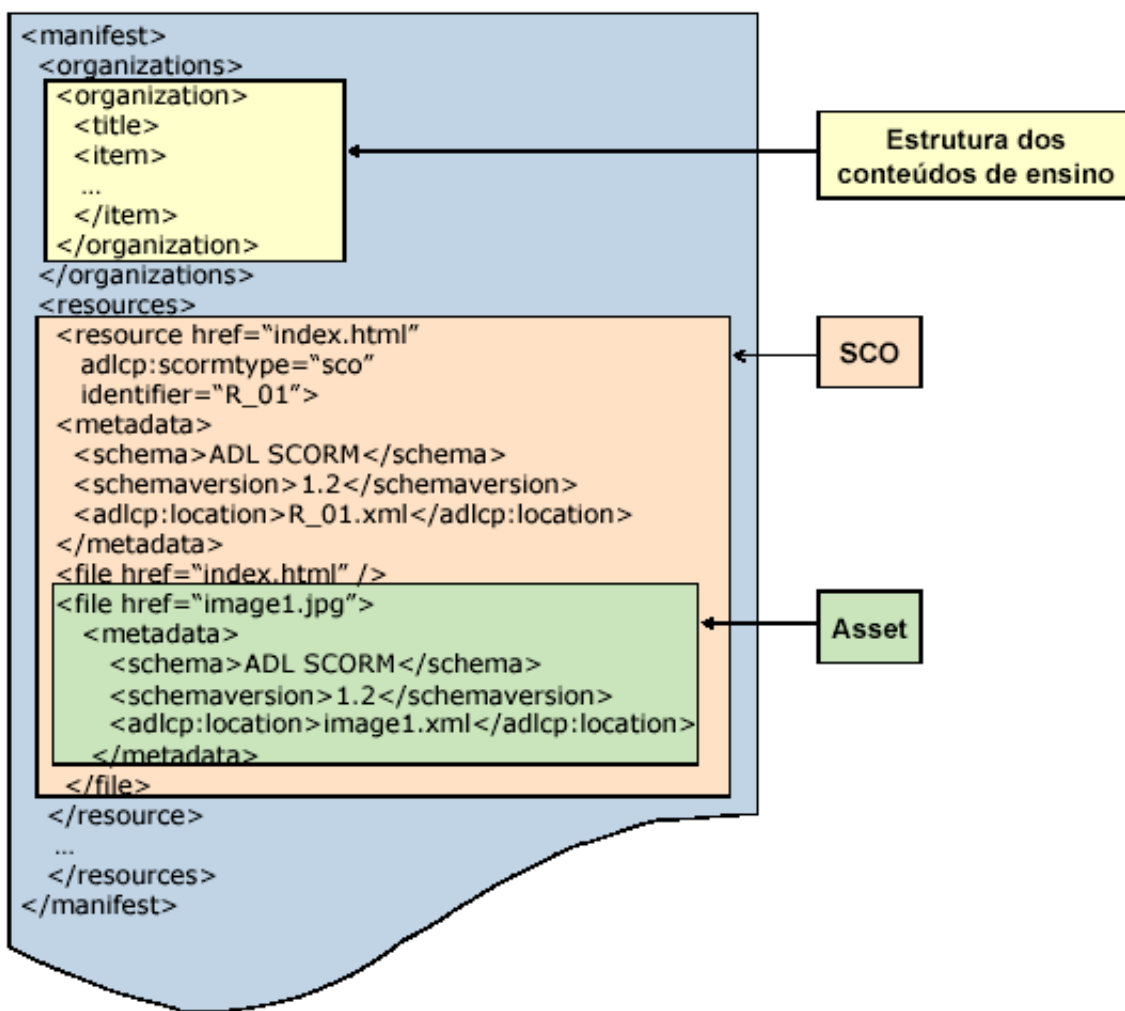


Figura 3.6: Exemplo de um imsmanifest.xml (extraída de [SCORM\_CAM]).

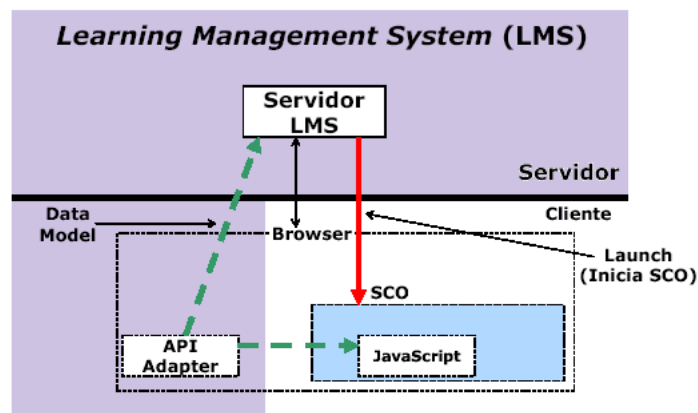
### 3.3 Run-Time Environment

A especificação do Run-Time Environment tem como objetivo permitir que os conteúdos de ensino possam ser visualizados em diferentes LMS e tenham em todos o mesmo comportamento. Para que isto seja possível, esta especificação determina a forma como os SCOs são enviados para o navegador e define o protocolo que os SCOs e o LMS irão usar para comunicar entre si.

O Run-Time Environment é constituído por três elementos que serão analisados nas seções seguintes:

- Mecanismo Launch: responsável por lançar o SCO para o browser e efetuar todas as diligências para que o SCO possa comunicar com o LMS.
- API (Application Program Interface): dispositivo que irá criar um canal de comunicação entre os SCOs e o LMS.
- Data Model: define a "linguagem" usada pelos SCOs e pelo LMS para comunicarem.

A figura mostra o papel destes três elementos e o modo como atuam numa plataforma de E-Learning:



**Figura 3.7:** Run-Time Environment de uma plataforma de E-Learning (extraída de [SCORM\_RTE]).

### 3.3.1 Launch

O Launch consiste num mecanismo, comum a todos os LMSs, responsável pelo envio dos recursos de ensino para o navegador do utilizador.

A idéia de criar de um mecanismo comum para lançar os recursos de ensino tem como objetivo obter um comportamento consistente em diferentes LMS's no ato de entrega dos recursos independentemente da forma como os LMS's estejam implementados.

Os componentes do Content Model que podem ser lançados pelo LMS são os Assets e os SCOs. Dependendo que se trate de um Asset ou de um SCO, o LMS desencadeia ações diferentes. Por exemplo, no caso de ser lançado um SCO, o LMS tem a responsabilidade de fornecer a estrutura que irá suportar a comunicação entre o SCO e o LMS, o API Adapter, descrito na seção seguinte.

Como foi já abordado, cabe ao LMS gerir a sequenciação e navegação entre diferentes recursos de ensino a partir da estrutura definida no Content Aggregation Model. O LMS baseia-se na interpretação de pré-requisitos presentes nos conteúdos de ensino para criar a sequência pela qual os conteúdos serão enviados para o utilizador.

Assim, a progressão através dos conteúdos de ensino, num determinado curso ou contexto de ensino, pode ser feita de diferentes maneiras conforme as capacidades do LMS e do modo como os recursos de ensino estejam construídos:

- Navegação sequencial: o aluno navega progressivamente pelos recursos de ensino.
- Navegação escolhida pelo aluno: o aluno escolhe a ordem pela qual quer aceder aos recursos de ensino, selecionando-os através de um menu com uma tabela de conteúdos.
- Navegação adaptada: o LMS determina quais os conteúdos de ensino e a ordem pela qual são entregues conforme o desempenho do aluno.

Depois de selecionado o recurso a ser lançado, o LMS localiza-o através do URL descrito no Content Package e envia-o para o navegador. O envio do recurso para o navegador é feito obrigatoriamente pelo protocolo HTTP.

segue uma descrição resumida do comportamento do LMS no Launch de Assets e SCOs:

#### **Assets**

No lançamento de Assets o único requisito a cumprir é que o Asset seja enviado para o navegador através de HTTP. Como o Asset não irá comunicar com o LMS não é necessário executar mais nenhuma operação.

### **SCOs**

Os SCO's são o meio privilegiado para divulgar os conteúdos de ensino pelo fato de serem dotados de capacidade de comunicação com o LMS. A capacidade de comunicação obriga contudo a alguns cuidados especiais para garantir um funcionamento correto do LMS e um acompanhamento eficiente da progressão do aluno. Assim, o LMS só pode lançar um SCO de cada vez e em cada momento apenas um SCO pode estar ativo. Por outro lado, um SCO só pode ser lançado pelo LMS. Um SCO não pode lançar outro SCO nem pode conter links que apontem para recursos externos ao próprio SCO.

Antes de enviar o SCO para o navegador, o LMS deve enviar uma janela principal com o API Adapter (Application Program Interface Adapter). O API Adapter é o responsável pelo estabelecimento da comunicação entre os SCO's e o LMS.

Depois de ser criada esta janela principal contendo o API Adapter, o SCO pode ser lançado para uma janela aberta pela janela principal ou para uma frame da janela principal, ou seja, o API Adapter deve estar colocado numa janela hierarquicamente superior à janela para onde o SCO vai ser lançado de modo a que este o possa encontrar. O API Adapter é sempre fornecido pelo LMS e pode ter implementações diferentes de LMS para LMS, embora respeite o mesmo Modelo de Dados.

Depois de lançado para o navegador, o SCO irá percorrer as janelas hierarquicamente superiores para encontrar o API fornecido pelo LMS. Assim que encontrar o API o SCO poderá iniciar a comunicação com o LMS.

## **3.3.2 API**

### **3.3.2.1 Descrição**

A Application Program Interface (API) é, como foi já descrito, o meio através do qual os SCOs e o LMS comunicam. A utilização da API pretende facilitar a interoperabilidade e reutilização dos conteúdos de ensino. A API permite que os SCOs comuniquem com o LMS de uma forma padrão, encapsulando os detalhes da comunicação do professor/formador que cria os conteúdos de ensino. De uma forma simples pode-se descrever uma API como sendo um conjunto predefinido de funções que o SCO tem à sua disposição.

O API Adapter consiste no software que implementa as funções da API. O API Adapter usa a capacidade de encapsulamento, inerente às linguagens orientadas a objetos, para ocultar os detalhes da implementação, permitindo assim que cada sistema de LMS crie o seu próprio API Adapter da forma mais conveniente tendo apenas que disponibilizar uma interface pública predefinida.

Todas as comunicações entre SCOs e o LMS são iniciadas pelo SCO. O LMS não dispõe de nenhum meio para chamar funções dentro de um SCO. Assim, depois do LMS lançar o SCO para o navegador invocando o mecanismo "Launch", cabe ao SCO localizar o API Adapter e iniciar a comunicação com o LMS para pedir e guardar os dados que pretender.

As funções disponibilizadas pelo API Adapter são oito, divididas em três categorias:

- Estado de Execução
  - LMSInitialize()
  - LMSFinish()
- Controle de Erros
  - LMSGetLastError()
  - LMSGetErrorString(errornumber)
  - LMSGetDiagnostic(parameter)
- Transferência de Dados
  - LMSGetValue(element)
  - LMSSetValue(element, value)
  - LMSCommit()

Segue-se uma descrição resumida de algumas das funções.

#### **LMSInitialize()**

Esta função indica ao LMS que o SCO pretende comunicar. Permite ao LMS "preparar-se" realizando as operações necessárias para processar os pedidos do SCO. O SCO deve executar obrigatoriamente o LMSInitialize antes de invocar qualquer outra função da API.

Retorna uma string com o valor *true* ou *false* caso a invocação tenha sido ou não bem sucedida.

### **LMSFinish()**

O SCO deve invocar esta função quando já não precisar comunicar com o LMS para que o LMS possa libertar os recursos gastos para atender aos pedidos do SCO.

Retorna uma string com o valor *true* ou *false* caso a invocação tenha sido ou não bem sucedida.

### **LMSGetLastError()**

Através da função `LMSGetLastError` o SCO pode saber se a chamada a alguma das funções da API descritas acima foi bem sucedida ou não e, neste caso, saber o que é que falhou. Sempre que é invocada a função retorna um código que representa o estado do erro gerado na última chamada à API.

Todas as funções da API (com exceção das funções do grupo de Controle de Erros: `LMSGetLastError`, `LMSGetErrorString` e `LMSGetDiagnostic`) geram um código de erro quando são chamadas. Este código assume o valor 0 se não ocorrer nenhum erro, ou o valor correspondente ao erro ocorrido.



A seguinte tabela mostra os códigos de erro mais comuns e o seu significado:

Tabela 3.1: Código dos erros mais frequentes

Tipo de Erro	Código	Descrição
	0	No Error
100's General errors	101	General exception
200's Syntax errors	201	Invalid argument
	202	Element cannot have children
	203	Element not an array. Cannot have count
300's LMS errors	301	Not initialized
400's Data model errors	401	Not implemented error
	402	Invalid set value, element is a keyword
	403	Element is read only
	404	Element is write only
	405	Incorrect Data Type

#### **LMSGetErrorString(errornumber)**

Esta função permite ao SCO obter uma descrição textual do erro ocorrido. Quando o SCO chama a função passando como argumento o código do erro para o qual pretende a descrição e a função devolve a string correspondente tal como está descrita na tabela acima. Exemplo:

```
var error = LMSGetErrorString("403");
```

O valor devolvido à variável error seria: "Element is read only"

#### **LMSGetDiagnostic(parameter)**

A função LMSGetDiagnostic permite aceder a descrições específicas de cada LMS para os erros ocorridos. Estas descrições serão à partida mais detalhadas que as fornecidas pela função GetErrorString porque são criadas dentro de cada LMS e permitem descrever o erro de forma mais pormenorizada e melhor integrada no contexto do LMS onde ocorreu o erro.

#### **LMSGetValue(element)**

Através desta função o SCO pode pedir dados ao LMS. O SCO especifica os dados

que pretende através do parâmetro passado como argumento. Os dados pedidos e a sintaxe dos parâmetros que são passados estão descritos no Data Model.

Todos os dados devolvidos pela função vêm no formato de strings.

#### **LMSSetValue(element, value)**

Com esta função o SCO pode enviar informação para o LMS. Conforme a implementação do API Adapter a informação pode ser enviada diretamente para o LMS ou guardada temporariamente numa memória cache e enviada mais tarde.

No segundo argumento da função é colocada a informação que se pretende guardar e no primeiro, indica-se em que elemento do modelo de dados ela deve ser guardada.

A função retorna *true* se não ocorrer nenhum erro ou *false* no caso contrário.

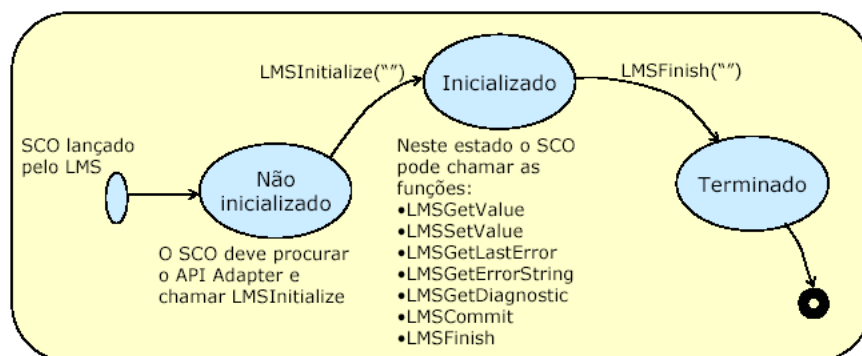
#### **LMSCommit()**

Esta função existe para garantir ao SCO que a informação enviada ao LMS através de uma chamada a LMSSetValue() será persistida. Nas implementações do API Adapter que usam memória cache, quando recebem a chamada a esta função devem enviar para o LMS todos os dados que tenham em memória e que ainda não tenham sido persistidos na base de dados do LMS. Se o API Adapter estiver implementado de forma a persistir os dados no fim de cada chamada a LMSSetValue, a função LMSCommit() torna-se redundante, devendo, no entanto, existir sempre para garantir conformidade total com o padrão SCORM. Nestes casos pode ser implementada como uma função oca, ou seja, não executa nenhuma operação e retorna sempre o valor *true*.

O parâmetro passado na chamada à função é sempre a string vazia e o valor de retorno será *true* ou *false* de acordo com o sucesso da chamada à função.

### 3.3.2.2 Interações dos SCOs e do LMS com o API Adapter

A figura abaixo representa um diagrama de estados de um SCO visto pelo API Adapter:



**Figura 3.8:** Diagrama de estados de um SCO visto pelo API Adapter (extraída de [SCORM\_RTE]).

Como se observa na figura, depois de ser lançado, o SCO deve localizar o API Adapter previamente enviado para o navegador pelo LMS. Para isso deve percorrer recursivamente as janelas-pai até o encontrar.

Todas as interações entre os SCOs e o API Adapter são realizadas através de ECMAScript (o standard de JavaScript). Apesar de o API Adapter poder ser implementado de diversas maneiras (Applets em Java, Flash, Remote Scripting, etc.) ele deve estar acessível através de JavaScript como sendo um objeto do tipo DOM (Document Object Model) com o nome API. Um SCO deve chamar sempre pelo menos as funções LMSInitialize() e LMSFinish().

### 3.3.3 Data Model

#### 3.3.3.1 Introdução

O Data Model foi criado para permitir que diferentes LMSs possam guardar e processar um conjunto definido de informações provenientes de qualquer SCO assim como, por outro lado, proporcionar aos SCO's uma forma padrão de aceder aos dados do LMS de que necessita. O Data Model consiste portanto, no estabelecimento de um protocolo que irá permitir a comunicação entre SCO's e LMS numa

"linguagem" que ambos entendem.

O Modelo de Dados utilizado pela SCORM deriva do modelo de dados criado pela AICC (Aviation Industry CBT Committee) denominado AICC CMI Data Model em que CMI é o acrónimo para Computer Managed Instructions. Em versões futuras do padrão poderão vir a ser incluídos outros modelos de dados, cuja estrutura será sempre semelhante à da AICC, prevendo-se portanto, uma transição fácil para os novos modelos quando estes forem adaptados.

### **3.3.3.2 Regras gerais do Modelo de Dados**

Antes de iniciar a descrição dos elementos que compõem o modelo de dados e da sua organização é oportuno fazer uma apresentação das regras gerais a observar na utilização do modelo de dados:

- O primeiro símbolo em cada elemento identifica o modelo de dados que está a ser utilizado. Esta regra permite que a mesma API funcione corretamente com modelos de dados diferentes.
- Existem três palavras chave reservadas que permitem obter informações acerca sobre o modelo de dados. Todas começam com o carácter "underscore" e são escritas em minúsculas:
  - `_version`: utilizada para determinar a versão do modelo de dados suportado pelo LMS
  - `_children`: utilizada para determinar os elementos contidos por um nó.
  - `_count`: utilizada para determinar o número de elementos contidos numa lista.
- Todas as listas começam em 0 e os elementos devem ser colocados nas listas de forma sequencial.
- O modelo de dados é implementado individualmente para cada SCO, isto é, cada SCO dispõe de campos específicos para guardar os seus dados e não pode acessar dados guardados por outro SCO.

### **3.3.3.3 Elementos do Modelo de Dados**

Uma vez que a lista de elementos que compõem o modelo de dados é bastante extensa e a sua compreensão é relativamente fácil, será feita apenas a apresentação

dos elementos mais relevantes do modelo de dados acompanhada de uma breve descrição.

**cmi.core**

Contém as informações mais importantes para os SCOs. O cmi.core é constituído pelos elementos: student\_id, student\_name, lesson\_location, credit, lesson\_status, entry, score, total\_time, lesson\_mode, exit e session\_time.

Se um SCO quiser saber os elementos do cmi.core suportados pelo LMS pode invocar a função LMSGetValue do seguinte modo:

```
var coreChildren = LMSGetValue("cmi.core._children");
```

Se o LMS suportar todas as funções previstas, irá responder com a string: "student\_id, student\_name, lesson\_location, credit, lesson\_status, entry, score, total\_time, lesson\_mode, exit, session\_time" contendo todos os elementos separados por vírgulas.

A tabela contém uma breve descrição de alguns destes campos:

Tabela 3.2: Descrição de alguns elementos de cmi.core

Função	descrição
cmi.core.student_id	Campo para guardar um código alfanumérico que identifica univocamente um determinado aluno. Corresponde ao número de aluno em qualquer instituição de ensino.
cmi.core.student_name	Campo onde é guardado o nome completo do aluno.
cmi.core.lesson_location	Corresponde ao ponto do SCO que estava sendo visualizado quando o aluno abandonou a sessão. Este valor é passado ao LMS pelo SCO quando o aluno sai do SCO e é depois recuperado quando o aluno visita novamente o SCO para permitir ao aluno recomeçar a lição no ponto em que a abandonou. A forma como o SCO guarda o ponto visitado diz respeito apenas ao SCO, o LMS limita-se a guardar essa informação.
cmi.core.lesson_status	Aqui é guardado o estado de cada SCO. Existem 6 valores possíveis: "passed", "completed", "failed", "incomplete", "browsed" e "not attempted".

continua na próxima página

Tabela 3.2: (continuação)

cmi.core.entry	Indica se o aluno já visitou o SCO. Existem três valores possíveis para este campo: "ab-initio" se o SCO nunca foi visitado, "resume" se o aluno já esteve no SCO ou , a string vazia, para outros casos como por exemplo se já tiver completado o SCO e decidiu vê-lo novamente.
cmi.core.score	Indicação do desempenho do aluno. Este campo possui três filhos: "raw", "min" e "max". (Só é utilizado nos SCOs em que se pretende avaliar o aluno como testes e exames.)
cmi.core.score.raw	Indica o resultado da avaliação do aluno no SCO. A forma de calcular e determinar o resultado da avaliação é decidida pelo formador que cria o SCO.
cmi.core.score.max	Resultado máximo que o aluno pode obter no SCO. Por exemplo, no caso de o SCO ser um exame, indica se está cotado para 20 valores, 100tem, no entanto que ser um número inteiro entre 0 e 100.
cmi.core.score.min	Nota mínima que pode ser atribuída ao aluno. Tem de ser um valor normalizado entre 0 e 100.
cmi.core.total_time	Tempo acumulado de todas as sessões do aluno no SCO. O formato deste campo deverá ser "HHHH:MM:SS.SS"

continua na próxima página

Tabela 3.2: (continuação)

cmi.core.exit	É uma indicação do motivo ou da forma como o aluno abandonou a sessão. Os valores possíveis para este campo são: "time-out", "suspend", "logout" ou (string nula) quando a saída é feita normalmente (completando o SCO).
cmi.core.session_time	Tempo gasto pelo aluno numa sessão, ou seja, o tempo decorrido entre o início e o fim de uma única sessão.

### **cmi.objectives**

O cmi.objectives guarda informações sobre o desempenho dos alunos relativamente aos objetivos propostos pelo SCO.

Este campo possui 3 elementos filhos: id, score e status. O cmi.objectives é um campo composto, isto é, permite guardar vários registos sob a forma de uma lista vetorial. Por exemplo, se um determinado SCO tiver definidos 3 objetivos, pode guardar o desempenho do aluno em cada um desses objetivos da seguinte forma:

```
LMSSetValue("cmi.objectives.0.status","failed");
LMSSetValue("cmi.objectives.1.status","passed");
LMSSetValue("cmi.objectives.2.status","incomplete");
```

Observe-se que as listas numeradas começam sempre em 0.

Se o SCO quiser num dado momento saber quantos registos dos objetivos existem no LMS pode usar a palavra chave "\_count":

```
var totalObj = LMSGetValue("cmi.objectives._count");
```



Segue-se na tabela a descrição dos elementos pertencentes a `cmi.objectives`. Os elementos descritos referem-se ao registro genérico 'n'.

Tabela 3.3: Descrição de alguns elementos de `cmi.objectives`

Função	descrição
<code>cmi.objectives.n.id</code>	Identificador do objetivo definido pelo formador que criou o SCO. Por exemplo: "objetivo 1", "P1", etc.
<code>cmi.objectives.n.score</code>	Semelhante ao campo <code>cmi.core.score</code> , este campo também possui os filhos: <code>raw</code> , <code>min</code> e <code>max</code> .
<code>cmi.objectives.n.status</code>	campo guarda o desempenho do aluno para o objetivo n. Existem 6 valores definidos para este campo: "passed", "completed", "failed", "incomplete", "browsed" e "not attempted".

### **cmi.interactions**

O campo `cmi.interactions` é também um campo composto onde são guardados os registos das interações dos alunos com o SCO. A cada registo deste campo corresponde uma interação do aluno com o SCO e para cada registo são guardados diversos detalhes dessa interação.

O `cmi.interactions` possui os filhos: `id`, `objectives`, `time`, `type`, `correct_responses`, `weighting`, `student_response`, `result` e `latency`.

Tabela 3.4: Descrição de alguns elementos de `cmi.interactions`

Função	descrição
<code>cmi.interactions.n.id</code>	Identificador da interação.
<code>cmi.interactions.n.time</code>	Indicador do tempo utilizado pelo aluno para completar a interação.

continua na próxima página

Tabela 3.4: (continuação)

<p>cmi.interactions.n.type</p>	<p>Identifica o tipo de interação que é guardada neste registro. O tipo de interação determina a forma como a resposta do aluno deve ser interpretada. Existem 8 valores pré-definidos para indicar o tipo de interação, embora possam ser usados outros valores:</p> <p><b>true-false:</b> questão apenas com duas respostas possíveis.</p> <p><b>choice:</b> questão com um número limitado de respostas que o aluno pode escolher. Podem estar corretas mais do que uma resposta.</p> <p><b>fill-in:</b> questão com uma resposta curta de apenas algumas palavras. O aluno tem de preencher um campo sem que exista uma resposta pré-definida.</p>
--------------------------------	--

continua na próxima página

Tabela 3.4: (continuação)

	<p><b>matching:</b> questão com dois conjuntos de itens em que cada item de um conjunto se relaciona com um ou mais itens do outro. A resposta à questão consiste em encontrar os pares de itens corretos.</p> <p><b>performance:</b> É uma questão semelhante à questão de resposta múltipla mas em que a resposta do aluno é dada através da realização de uma ação ou tarefa.</p> <p><b>likert:</b> oferece ao aluno um grupo de alternativas dentro de uma série contínua. A resposta é geralmente baseada na opinião do aluno.</p> <p><b>numeric:</b> a resposta à pergunta é um valor numérico, não necessariamente inteiro.</p>
<p>cmi.interactions.n.correct_responses.n.pattern</p>	<p>Descreve as respostas possíveis à interação n. Pode existir mais do que uma resposta correta, assim como podem existir respostas mais corretas do que outras. avaliação.</p>

continua na próxima página

Tabela 3.4: (continuação)

cmi.interactions.n.weighting	Consiste num campo numérico que atribui um peso diferente a cada interação. A finalidade deste campo é permitir avaliar a importância relativa entre interações. Deste modo, se uma interação possuir um peso de 25 e outra tiver peso 15, o resultado da primeira terá mais influência no resultado final da avaliação.
------------------------------	--

### 3.4 Sequencing and Navigation

O sequenciamento, como o próprio nome indica, cuida do comportamento do curso, definindo os requerimentos e limitações de cada objeto de aprendizado e permitindo criar um encadeamento lógico entre os mesmos de modo que todos usuários possam executar o curso de uma maneira consistente.

A navegação, por outro lado, permite que eventos de navegação sejam iniciados pelos próprios objetos de aprendizado em tempo de execução com base no próprio modelo de sequenciamento definido para o curso.

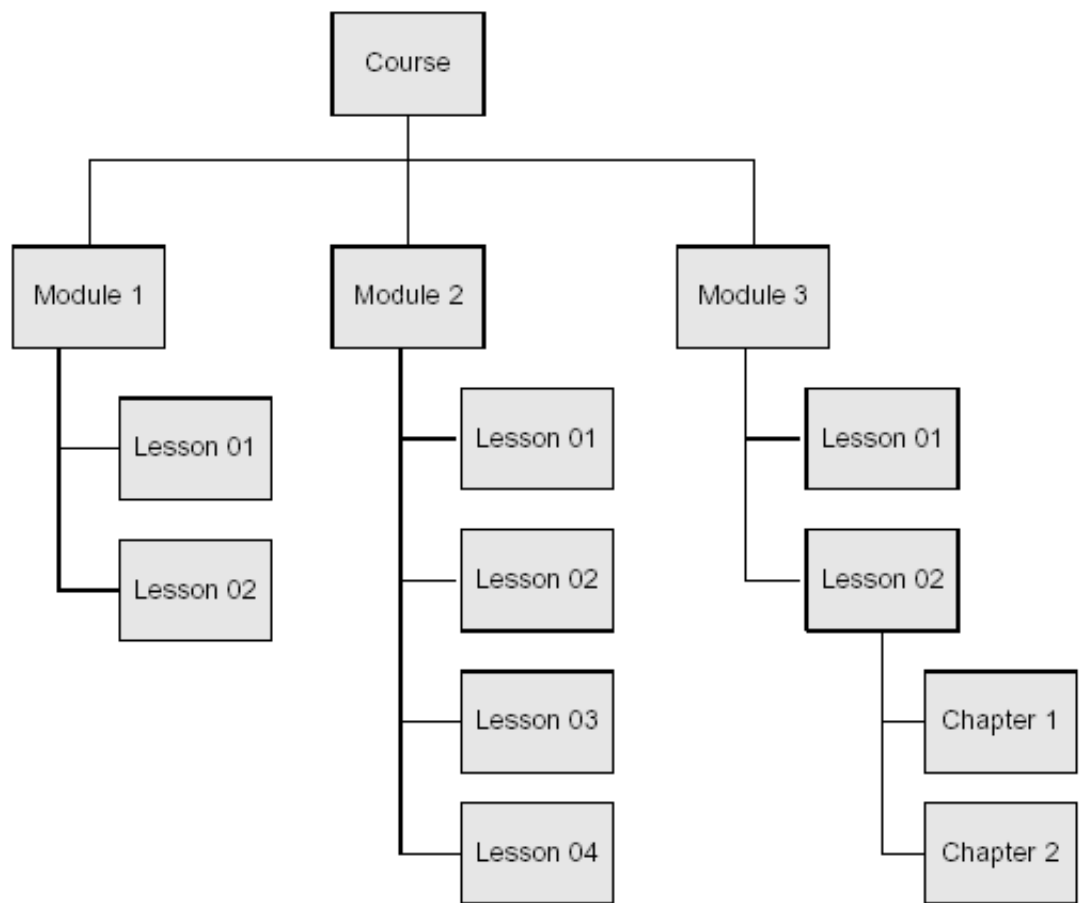
Algumas estruturas são definidas para garantir uma seqüência adequada ao curso e que os SCO's se comuniquem com o LMS e iniciem eventos de navegação de forma mais independente.

- Activity Tree: Árvore de atividades, descreve a estrutura das atividades.
- Cluster: definem agrupamentos de atividades de aprendizado.

#### 3.4.1 Activity Tree

A Activity Tree permite que se descreva exigências de processamento e informações assim como algoritmos de sequenciamento e comportamento, independente da forma de implementação adotada. A figura 3.9 é um exemplo de Activity Tree.

A raiz da Activity Tree é "Curso"propriamente dito, mas é importante notar que ela também é uma atividade. Outra observação importante é que o padrão

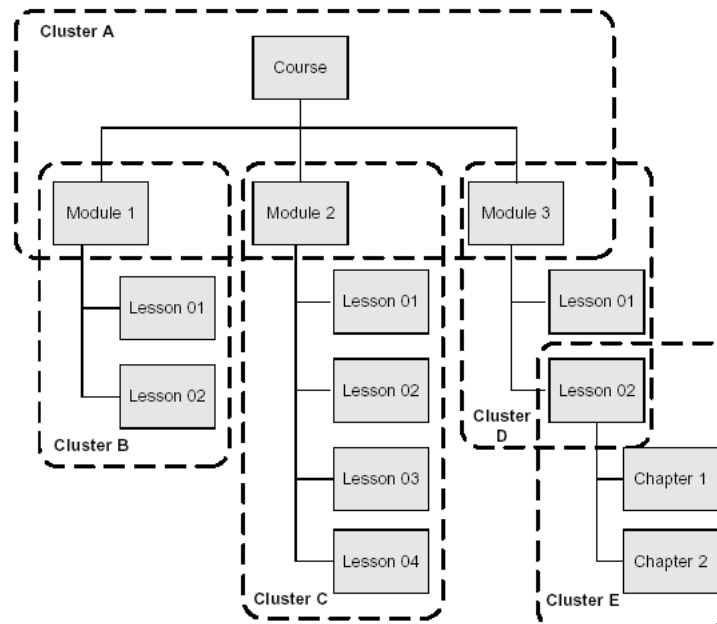


**Figura 3.9:** Exemplo de uma Activity Tree (extraída de [SCORM\_SN]).

SCORM não a define como uma estrutura estática, portanto é possível criá-la dinamicamente.

### 3.4.2 Cluster

Um cluster (Agrupamento) é uma forma específica de atividade de aprendizado que possui sub-atividades. Um cluster possui uma única atividade pai e seus filhos imediatos, não pertencendo ao cluster os filhos de seus filhos. Um cluster é um ajuntamento de folhas de atividade, sendo que uma folha de atividades apenas não caracteriza um cluster.



**Figura 3.10:** Exemplo de Cluster (extraída de [SCORM\_SN]).

Na figura 3.10 nota-se a existência de cinco clusters, cada cluster é definido por um retângulo pontilhado, o cluster A, possui somente quatro atividades: a atividade pai "Curso" e as atividades filhas que são os módulos (clusters B, C e D). Estes por sua vez possuem atividades filhas que são as "Lições". Exceto na "Lição 2" do "Módulo 3" que é um novo cluster constituído das atividades "Capítulos".

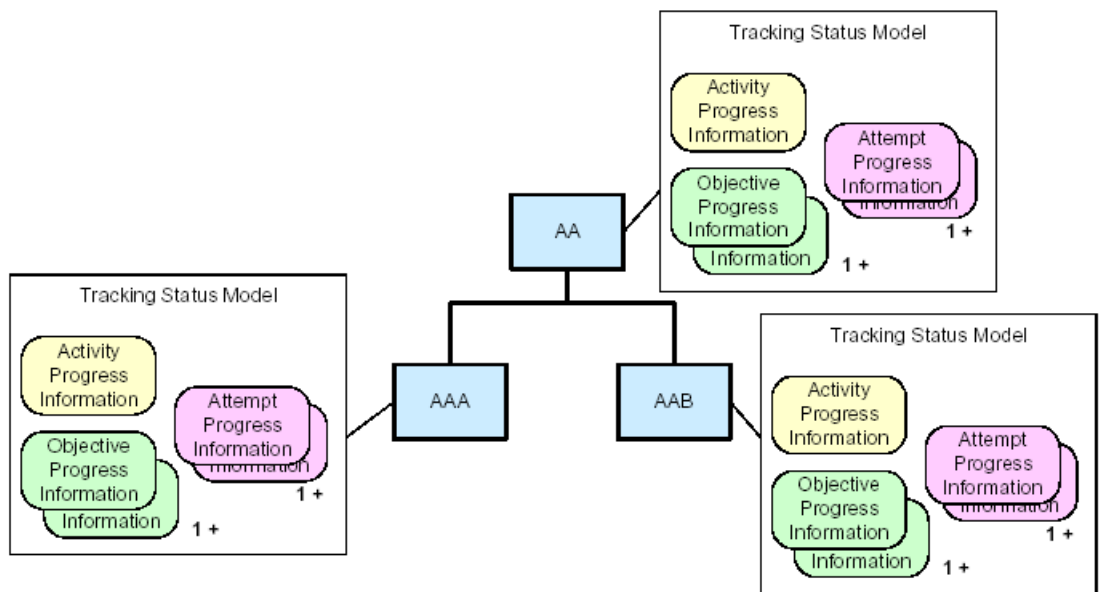
No contexto do SCORM, a experiência de aprendizado de um estudante é proveniente das atividades da Activity Tree realizadas pelo estudante descrita na seção 3.4.1. A navegação é o processo pelo qual LMS recebe os pedidos do estudante por atividades de aprendizado, ou seja, as requisições de um SCO.

### **3.5 Tracking Model**

Nas versões anteriores do SCORM o único modelo de dados especificados era Data Model. Suas informações eram utilizadas para rastrear as interações de um estudante com o SCO. Na versão 1.3 o SCORM define um novo modelo de dados adicional, o Tracking Model (Modelo de Localização), ele é uma coleção dinâmica de informações associadas a um estado de seqüência, para cada atividade de aprendizado, para cada estudante. Valores iniciais serão definidos para todo o modelo, durante as experiências de aprendizado os seus elementos serão atualizados para refletir o atual estado do aluno.

Todas as atividades têm associadas a si uma informação específica sobre o estado "status" de um estudante em uma determinada atividade. Um exemplo de Activity Tree associada a uma informação de "status" é mostrado na figura 3.11, Através das requisições feitas pelo SCO, o LMS atualiza sempre estas informações permitindo que acompanhe sempre o estado atual de estudante.





**Figura 3.11:** Tracking Model (extraída de [SCORM\_SN]).



## Capítulo 4

# Conclusão

### 4.1 Considerações Finais

Como foi apresentado, a padronização na criação de sistemas gerenciadores de aprendizagem, se mostrou muito importante uma vez que o mercado estava sendo inundado por uma grande oferta de LMS's criados a partir de tecnologias diferentes. Diante dessa configuração, vários consórcios começaram a criar padrões que pudessem gerir a criação desses LMS's.

O padrão SCORM, se destacou justamente por não ser mais um padrão, mais sim por surgir como sendo um conjunto de outros padrões. O SCORM faz uso de várias diretrizes de outros padrões, tornando-se assim, mais abrangente.

O SCORM foi e ainda está sendo amplamente absorvido pelo mercado, é difícil encontrar um LMS que não de um suporte mínimo a ele, hoje podemos considerar o SCORM um padrão que já está consolidado, mas que tem como característica buscar uma constante evolução.

### 4.2 Sugestões para Trabalhos Futuros

Para trabalhos futuros, pode-se sugerir a implementação de um protótipo de um sistema gerenciador de aprendizagem que suporte SCORM, e posteriormente a consolidação de um LMS adequado a um padrão que garante alto nível mundialmente levando o nome do departamento de ciência da computação da UFLA \_ Universidade Federal de Lavras.



# Referências Bibliográficas

- [ADL] ADL *Advanced Distributed Learning*. <http://www.adlnet.org/>
- [IEEE] IEEE *IEEE Information Technology Learning Technology Learning Objects Meta-data*. LOM: Working Draft v6.1 Resource Meta-data Specification Version 1.2. Disponível em: <http://ltsc.ieee.org/>
- [IMS] IMS *IMS Learning Resource Meta-data Specification Version 1.2*. Disponível em: <http://www.imsglobal.org/>
- [E-Learning Brasil] E-Learning Brasil *E-Learning Brasil*. <http://www.elearningbrasil.com.br/>
- [ABED] ABED *Associação Brasileira de educação a Distância*. <http://www.abed.org.br/>
- [Moodle] Moodle *Modular Object-Oriented Dynamic Learning Environment*. <http://moodle.org/>
- [AICC] AICC *Aviation Industry CBT Committee*. <http://www.aicc.org/>
- [ATutor] ATutor *ATutor Learning Content Management System*. <http://www.atutor.ca/>
- [WEBAULA] WebAula *Zargon e Poliedro*. <http://www.webaula.com.br/>

- [NUNES1992] NUNES, Ivônio Barros *Noções de Educação a Distância*(1992.) Disponível em: [http://www.intelecto.net/ead\\_textos/ivonio1.html](http://www.intelecto.net/ead_textos/ivonio1.html)
- [MOORE1996] MOORE, Michel G., KEARSLEY, Greg. *Distance education: a systems view*. Belmont (USA) : Wadsworth Publishing Company (1996.)
- [ARETIO1994] ARETIO, Lorenzo Garcia *Educación a distancia hoy*. Madrid: UNED. Educación a distancia(1994).
- [KEEGAN1991] KEEGAN, S. D, HOLMBERG, B., MOORE, M. G., *Distance education International perspectives*. London: Routledge, (1991.)
- [PRETI1996] PRETI, Oreste. Educação a distância: uma prática educativa mediadora e mediatizada. In: PRETI, Oreste. Educação a distância: inícios e indícios de um percurso. Cuiabá: NEAD/IE - UFMT (1996.)
- [GNU] GNU *General Public License*. <http://www.gnu.org/copyleft/gpl.html>
- [SCORM\_CAM] *Sharable Content Object Reference Model(SCORM)Content Aggregation Model (CAM) Version 1.3.1* disponível em <http://www.adlnet.org/>
- [SCORM\_RTE] *Sharable Content Object Reference Model(SCORM)Run-Time Environment (RTE) Version 1.3.1* disponível em <http://www.adlnet.org/>
- [SCORM\_SN] *Sharable Content Object Reference Model(SCORM)Sequencing and Navigation (SN) Version 1.3.1* disponível em <http://www.adlnet.org/>