



**Carlos Alberto B.  
Sousa Pinto**

**Arquitectura para Gestão de Actividades em  
Plataformas Distribuídas de e-Learning**



**Carlos Alberto B.  
Sousa Pinto**

**Arquitectura para Gestão de Actividades em  
Plataformas Distribuídas de e-Learning**

tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Ciências e Tecnologia da Comunicação, realizada sob a orientação científica do Dr. Fernando Manuel dos Santos Ramos, Professor Catedrático do Departamento de Comunicação e Arte da Universidade de Aveiro

Dedico este trabalho, a título póstumo, ao Professor Doutor Altamiro Barbosa Machado, meu mestre e amigo.

## **o júri**

presidente

**Prof. Dr. Joaquim José Borges Gouveia**  
professor catedrático da Universidade de Aveiro

**Prof. Dr. Fernando Manuel dos Santos Ramos**  
professor catedrático da Universidade de Aveiro

**Prof. Dr. Luís Alfredo Martins do Amaral**  
professor associado da Escola de Engenharia da Universidade do Minho

**Prof. Dr. Óscar Emanuel Chaves Mealha**  
professor associado da Universidade de Aveiro

**Prof. Dr. António José Nunes Mendes**  
professor auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

**Prof. Dr. Álvaro Pedro de Barros Borges Reis Figueira**  
professor auxiliar da Faculdade de Ciências da Universidade do Porto

**Prof. Dr. Osvaldo Arede Santos**  
professor adjunto da Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco

**Prof. Dra. Ângela Carrancho da Silva**  
professora adjunta da Faculdade de Educação da Universidade do Estado do Rio de Janeiro  
professora e pesquisadora do Programa de Pós-graduação da Fundação CESGRANRIO

## **agradecimentos**

Este trabalho só se tornou possível porque muitas pessoas, de uma forma ou de outra, me ajudaram a ultrapassar todas as dificuldades por que fui passando. Com o receio de esquecer alguma delas, manifesto o meu agradecimento a todos quantos, com a sua presença, companhia e amizade, me ajudaram na realização desta tarefa.

Porque mais de perto acompanharam esta minha caminhada, quero deixar aqui o meu especial agradecimento:

Ao Professor Doutor Fernando Ramos, orientador deste trabalho, que em condições especiais aceitou essa tarefa, a quem reconheço a competência na orientação, por diversas vezes constatada, e cujo apoio foi muito para além do âmbito deste trabalho.

Ao Departamento de Sistemas de Informação da Universidade do Minho, nas pessoas do Professor Doutor João Álvaro Carvalho e do Professor Doutor Luis Amaral por, na qualidade de directores do departamento durante o tempo da realização do trabalho, sempre terem proporcionado as condições necessárias à realização do mesmo.

Aos meus colegas do DSI, em especial à Maribel, à Ana Alice, ao Pedro Coutinho e ao Leonel, pelas diversas formas de ajuda que me proporcionaram e à amizade que nunca regatearam.

Ao Paulo e à Jú, pelo apoio, pela ajuda e pela amizade.

Ao Professor Fernandes Almeida, pela amizade, pelo incentivo e por, apesar de já não estar ao serviço, ainda ter arranjado tempo para me ensinar o significado do termo “despoletar”.

Ao Chico e à Elda, pela amizade e pelos almoços, que, embora tendo diminuído em número, não perderam a importância que sempre tiveram na consolidação de uma amizade já antiga.

À Dra. Clara Pinho, médica competente, por me ter ajudado a regressar.

À Idalina, minha companheira, por ter caminhado comigo nos últimos anos e ter aguentado a viagem.

Ao meu pai, por me ter transmitido as capacidades de trabalho e de sacrifício, sem as quais este trabalho nunca teria chegado ao fim.

## palavras-chave

e-Learning, Metadados, SCORM, Gestão de Actividades.

## resumo

O desenvolvimento de LMSs bem como dos conteúdos a serem utilizados nos cursos que os mesmos suportam, tem sido feito pelos produtores de software e de conteúdos, sem atender a quaisquer normas, fazendo com que não seja possível a compatibilização dos diferentes sistemas existentes.

Preocupações de reuso e interoperabilidade estão na base de diversos projectos de normalização nesta área, levados a cabo por organizações e consórcios internacionais, envolvendo organismos governamentais, empresariais e académicos de todo o mundo.

Até ao momento, os principais esforços têm sido canalizados para a questão da construção de unidades de aprendizagem (conteúdos) que possam ser reutilizáveis, estando em curso trabalhos relacionados com a normalização dos LMSs, por forma a que estes possam gerar, utilizar e trocar entre si, conteúdos com as mais diversas amplitudes, indo do simples texto até ao curso completo.

Algumas das organizações e consórcios que mais se têm evidenciado neste tipo de trabalho incluem a ADL (Advanced Distributed Learning), o IMS Global Learning Consortium, Inc., o AICC (Aviation Industry Computer Based Training Committee), e o IEEE LTSC (Institute of Electrical and Electronic Engineers Learning Technologies Standard Committee), entre outros.

O projecto SCORM (Sharable Content Object Reference Model) da ADL é talvez aquele que neste momento melhor representa o esforço que tem sido feito no sentido da normalização referida.

Contudo, até à data, nenhum dos projectos conhecidos se preocupou com questões de gestão operacional dos cursos, deixando por cobrir aspectos achados importantes para o aumento da probabilidade de sucesso dos processos de ensino/aprendizagem.

A presente tese pretende descrever o trabalho de obtenção de um referencial para a camada de gestão on-line dos processos de e-learning suportados por LMSs. Tal camada é proposta de forma a poder integrar-se no todo que constitui uma plataforma de ensino/aprendizagem desenvolvida de acordo com as recomendações SCORM e, para além da proposta de uma camada de metadados de gestão, apresenta a forma como a referida integração poderá ser realizada.

**keywords**

e-Learning, Metadata, SCORM, Activity Management.

**abstract**

The development of the LMSs as well as the development of the contents to be used inside the courses that the LMSs support has been done without any standard guidelines. This fact addresses the problem of the existence of non compatible systems in the e-learning market. So, interoperability and reuse are the concerns of several organizations and consortia involving governmental and corporate members.

Till now, the main efforts of those organizations and consortia are focused on the building and delivering of contents that could be reused. Simultaneously there are some works running towards a standardized process of LMSs development, so that it could be possible to create courses and generate, use and interchange contents among them. Those contents could have different levels of granularity, beginning on a simple text and ending in a complete course.

Some of the most important organizations and consortia working in the area are ADL (Advanced Distributed Learning), IMS Global Learning Consortium, Inc, AICC (Aviation Industry Computer Based Training Committee, IEEE LTSC (Institute of Electrical and Electronic Engineers Learning Technologies Standard Committee) and others.

The ADL SCORM (Sharable Content Object Reference Model) project is perhaps the one that, at this moment, better represents the efforts made in the standardization works referred. However, till today, there isn't any known project covering the on-line management of activities execution during a course operationalization.

We believe that such such management capabilities could allow better results in what concerns the success of the teaching/learning process.

This thesis describes the work done towards a proposal of a referential for an on-line management layer to be integrated in the global architecture of a LMS.

Doing this work, we had in mind the need of integration of the proposed new layer with the existent SCORM recommendations for the LMSs development. The integration process and the management metadata are also presented and described.

<b>1 Introdução</b> .....	1
1.1 Contexto .....	3
1.2 Gestão de actividades em cursos suportados por LMS .....	4
1.2.1 Antecedentes .....	4
1.2.2 Questões de investigação e objectivos .....	7
1.3 Metodologia do trabalho .....	8
1.4 Organização da Tese .....	9
<b>2 Enquadramento do Trabalho</b> .....	11
2.1 Desenvolvimento de normas .....	12
2.2 Características dos LMS existentes no mercado .....	13
2.3 Motivações para a realização do trabalho .....	14
<b>3 Principais Projectos de Normalização</b> .....	17
3.1 IEEE LTSC .....	18
3.1.1 Introdução.....	18
3.1.2 Objectivos.....	18
3.1.3 Normas, especificações ou recomendações geradas .....	19
3.2 O Projecto IMS Global Learning Consortium .....	26
3.2.1 Introdução .....	26
3.2.2 Objectivos .....	28
3.2.3 Normas, especificações ou recomendações geradas .....	28
3.3 O Projecto AICC.....	46
3.3.1 Introdução .....	46
3.3.2 Objectivos.....	48
3.3.3 Normas, especificações ou recomendações geradas .....	48
3.4 O Projecto ARIADNE .....	51
3.4.1 Introdução .....	51
3.4.2 Objectivos .....	52
3.5 O Projecto ADL SCORM.....	53
3.5.1 Introdução .....	53
3.5.2 Objectivos .....	55



3.5.3 Normas, especificações ou recomendações geradas .....	55
<b>4 Técnicas de Captura e Representação de Conhecimento Utilizadas .....</b>	<b>66</b>
4.1 Introdução .....	66
4.2 Modelo Entidades-Relacionamentos.....	66
4.3 Ontologias de Domínio .....	67
4.4 A Linguagem UML .....	73
<b>5 Estudo, proposta e discussão de uma camada de gestão de actividades em tempo real.....</b>	<b>77</b>
5.1 Captura de Conhecimento e Modelação do Sistema .....	77
5.1.1 Introdução .....	77
5.1.2 Usando UML para Modelar as Interações com um LMS .....	78
5.1.3 Outras Representações complementares .....	94
5.1.3.1 Estrutura hierárquica de navegação .....	94
5.1.3.2 Modelo conceptual em camadas .....	96
5.2 O problema ao nível da gestão das actividades .....	97
5.2.1 Introdução .....	97
5.2.2 Tipos de Actividades .....	100
5.2.2.1 Actividades sequenciais .....	100
5.2.2.2 Actividades paralelas .....	101
5.2.2.3 Actividades de execução aleatória .....	102
5.2.2.4 Actividades compostas de subactividades .....	103
5.2.2.5 Actividades condicionadas .....	106
5.2.2.6 Actividade de constituição de grupos .....	107
5.2.3 Operações Sobre Actividades .....	108
5.2.3.1 Alteração da ordem das actividades .....	108
5.2.3.2 Execução de actividades por grupos.....	109
5.2.3.3 Execução de actividades ao longo de diversas sessões	111
5.2.3.4 Execução aleatória de actividades .....	111
5.2.3.5 Execução condicionada de actividades .....	112
5.2.3.6 Execução de actividades de recuperação .....	113
5.2.4 A Componente Notificações .....	114

5.2.4.1	Tipos de notificações inicialmente previstos .....	114
5.2.4.2	Notificações do tipo informação .....	115
5.2.5	Os diferentes actores previstos .....	116
5.2.6	O modelo de gestão de actividades melhorado .....	117
5.2.7	Regras na definição de identificadores .....	119
5.3	Funcionalidades da componente software .....	120
5.4	Validação do Modelo – Questões de Competência .....	125
5.5	Arquitectura Proposta .....	128
5.5.1	Metadados de Gestão .....	130
5.5.2	Pontos de ancoragem ao projecto SCORM .....	159
5.6	Contribuições para o projecto SCORM .....	163
5.7	Validação dos resultados .....	164
<b>6</b>	<b>Conclusões</b> .....	<b>167</b>
6.1	Respondendo às questões de investigação .....	167
6.2	Quanto às Técnicas de modelação utilizadas .....	169
6.3	Dificuldades e limitações sentidas .....	172
6.4	Trabalho futuro .....	172
	<b>Referências</b> .....	<b>175</b>

## Lista de Acrónimos

ADL	-	Advanced Distributed Learning
AGR	-	AICC Guidelines & Recommendations
AICC	-	Aviation Industry Computer Based Training Committee
ANSI	-	American National Standards Institute
API	-	Application Program Interface
ARIADNE	-	Alliance of Remote Instructional Authoring and Distribution Networks for Europe
ATA	-	Airline Transport Association
BNF	-	Bachus-Naur Form
CAM	-	Content Agregation Model
CBT	-	Computer Based Training
CD-ROM	-	Compact Disk-Read Only Memory
CEN	-	European Committee for Standardization
CMI	-	Computer Mediated Instruction
CO	-	Content Object
DCMI	-	Dublin Core Metadata Initiative
DoD	-	Department of Defense
DREL	-	Digital Rights Expression Languages
DRM	-	Digital Rights Management
DVD	-	Digital Versatile Disk
DVD-ROM	-	Digital Versatile Disk-Read Only Memory
ECMA	-	European Computer Manufacturers Association
Eduspecs	-	Industry Canada's EduSpecs Initiative
eLearning	-	Electronic Learning
ELI	-	EDUCAUSE Learning Initiative
eLMS	-	Electronic Learning Management System
EML	-	Educational Modelling Language
eXe	-	eLearning XHTML editor
FCT	-	Fundação Para a Ciência e Tecnologia
IEC	-	International Electrotechnical Commission
IEC	-	International Electrotechnical Commission
IEEE	-	Institute of Electrical and Electronic Engineers

IMS	-	Instructional Management System
IMS LD	-	IMS Learning Design
IMS LIP	-	IMS Learner Information Package
IMS SS	-	IMS Simple Sequencing
ISO	-	International Standards Organization
ISSS	-	Information Society Standardization System
LAMS	-	Learning Activity Management System
LAN	-	Local Area Network
LIP	-	Learner Information Package
LIS	-	Learner Information Systems
LMS	-	Learning Management System
LO	-	Learning Object
LOM	-	Learning Object Metadata
LTS	-	Learning Technology System
LTSC	-	Learning Technology Standard Committee
MOODLE	-	Modular Object-Oriented Dynamic Learning Environment
MP3	-	MPEG Audio Layer-3
NLII	-	National Learning Infrastructure Initiative
ODL	-	Open and Distance Learning
OMG	-	Object Management Group
OO	-	Orientado a Objectos (Object Oriented)
PENS	-	Package Exchange Notification Services
PT	-	Portugal Telecom
QTI	-	Question & Test Interoperability
R&D	-	Research and Development
RDCEO	-	Reusable Definition of Competency or Educational Objective Specification
RDF	-	Resource Description Framework
RELOAD	-	Reusable eLearning Object Authoring and Delivery
RLI	-	Resource List Interoperability
RTE	-	Run-Time Environment
RTS	-	Run-Time Service
SCO	-	Sharable Content Object
SCORM	-	Sharable Content Object Reference Model

SN	- Sequencing and Navigation
TI	- Tecnologias da Informação
TIC	- Tecnologias da Informação e da Comunicação
UA	- Universidade de Aveiro
UML	- Unified Modeling Language
UoL	- Unit of Learning
VDEX	- Vocabulary Definition Exchange
VLE	- Virtual Learning Environment
W3C	- The World Wide Web Consortium
WBT	- Web Based Training
WG	- Work Group
XMI	- XML Metadata Interchange
XML	- Extensible Markup Language

<b>Índice de Figuras</b>	<b>Pág.</b>
<b>Figura 1</b> - Organização da norma SCORM com base em uma coleção de livros. Adaptado de (ADL, 2004a).	- 4
<b>Figura 2</b> - Unidade atômica de gestão de actividades em tempo real.	- 5
<b>Figura 3</b> - Modelo genérico de um LMS. Adaptado de (ADL, 2004a)	- 14
<b>Figura 4</b> - Agregação semântica do Learning Design no nível C. Adaptado de IMS Learning Design Information Model (IMS,2003a).	- 40
<b>Figura 5</b> - Modelo conceptual do Learning Design. Adaptado de IMS Learning Design Information Model (IMS, 2003a)	- 42
<b>Figura 6</b> - Figura 6 – Evolução do projecto SCORM. Adaptado de (ADL, 2004a).	- 54
<b>Figura 7</b> - Representação conceptual do Run-Time Environment SCORM. Adaptado de (ADL, 2004c).	- 58
<b>Figura 8</b> - Modelo temporal de relacionamentos envolvendo um SCO. Adaptado de (ADL, 2004c).	- 60
<b>Figura 9</b> - Relação entre a Organização de Conteúdos (manifesto) e a Árvore de Actividades. Adaptado de (ADL, 2004d).	- 62
<b>Figura 10</b> - Exemplo de uma Árvore de Actividades derivada a partir de um manifesto com submanifesto. Adaptado de (ADL, 2004d).	- 63
<b>Figura 11</b> - Exemplo de Clusters. Adaptado de (ADL, 2004d).	- 64
<b>Figura 12</b> - Vista parcial da ontologia relativa à gestão de cursos na Web (obtida através da ferramenta Protégé2000).	- 71
<b>Figura 13</b> - Metadados referentes à ontologia de gestão de cursos na Web (Obtida através da ferramenta Protégé2000).	- 72
<b>Figura 14</b> - Classificação dos tipos de diagramas UML. Adaptado de UML Distilled Third Edition (Fowler, 2004).	- 74
<b>Figura 15</b> - Diagrama de Caso de Uso “Atender Pedido de Inscrição em Curso_v2”.	- 79
<b>Figura 16</b> - Diagrama de Caso de Uso “{U0}LMS” (Nível Zero).	- 81
<b>Figura 17</b> - Diagrama de Caso de Uso {U0.1}Atender Pedido de Conta.	- 83
<b>Figura 18</b> - Diagrama de Sequência do Diagrama de Casos de Uso {U0.1} Atender Pedido de Conta.	- 83
<b>Figura 19</b> - Diagrama de Caso de Uso {U0.3}Emitir Notificações.	- 86

<b>Figura 20</b> - Diagrama de Sequência do Diagrama de Casos de Uso {U0.3} Emitir Notificações.	- 87
<b>Figura 21</b> - Diagrama de Caso de Uso {U0.4} Atender Pedido de Inscrição em Curso.	- 88
<b>Figura 22</b> - Diagrama de Sequência do Diagrama de Casos de Uso {U0.4} Atender Pedido de Inscrição em Curso.	- 89
<b>Figura 23</b> - Diagrama de Caso de Uso {U0.5} Assistir Criação de Novo Curso.	- 90
<b>Figura 24</b> - Diagrama de Sequência do Diagrama de Casos de Uso {U0.5} Assistir Criação de Novo Curso.	- 91
<b>Figura 25</b> - Diagrama de Caso de Uso {U0.8} Suportar Execução de Actividades.	- 92
<b>Figura 26</b> - Diagrama de Sequência do Diagrama de Casos de Uso {U0.8} Suportar Execução de Actividades.	- 93
<b>Figura 27</b> - Estrutura hierárquica de navegação num LMS.	- 95
<b>Figura 28</b> - Representação conceptual em camadas das interacções com um LMS.	- 96
<b>Figura 29</b> - Invocação de uma estrutura com duas subactividades a partir de uma actividade de nível superior (actividade pai).	- 104
<b>Figura 30</b> - Recuperação de actividade não executada.	- 113
<b>Figura 31</b> - Modelo melhorado da unidade atómica de gestão de actividades, em tempo real.	- 118
<b>Figura 32</b> - Vista parcial do esquema de base de dados de suporte a um LMS.	- 127
<b>Figura 33</b> - Arquitectura de relacionamento da camada de gestão com o restante LMS e das interacções dos diferentes actores com o sistema global.	- 129
<b>Figura 34</b> - Visão gráfica do esquema XML para a camada de gestão, obtido via Turbo XML da TIBCO.	- 131
<b>Figura 35</b> - Representação gráfica resumida do esquema XML relativo a um Curso.	- 133
<b>Figura 36</b> - Subesquema XML para a componente Actividade.	- 134
<b>Figura 37</b> - Subesquema XML para a componente Evento.	- 134
<b>Figura 38</b> - Subesquema XML para a componente Actor.	- 135
<b>Figura 39</b> - Subesquema XML para a componente Grupo.	- 135

<b>Índice de Tabelas</b>	<b>Pág.</b>
<b>Tabela 1</b> - Excerto da tabela de actividades identificadas no projecto ODL Toolbox Obtida a partir do relatório final do projecto (Ramos et al, 2003).	- 5
<b>Tabela 2</b> - Excerto da tabela com a definição dos elementos de metadados identificados no projecto ODL Toolbox. Obtida a partir do relatório final do projecto. (Ramos et al, 2003).	- 7
<b>Tabela 3</b> - Fases associadas a uma actividade e tipos de notificações correspondentes.	- 118
<b>Tabela 4</b> - Metadados associados à definição de um LMS	- 137
<b>Tabela 5</b> - Elementos do modelo de dados do RTE. Adaptado de (ADL, 2004c)	- 160



# Convenções de escrita

Por uma questão de simplificação foram adoptadas nesta tese algumas convenções de escrita que se apresentam de seguida:

1. Palavras ou expressões em língua estrangeira, para as quais não existam termos correspondentes aceites em Português, são apresentadas em *itálico*.
2. Em alguns casos, desde que tal não comprometa o entendimento do texto e se esteja em presença de termos universalmente reconhecidos e aceites, poder-se-á utilizar o termo em língua estrangeira, mesmo havendo termos em Português equivalentes.

Exemplos:

- draft com o mesmo significado de provisório;
- e-mail com o mesmo significado de correio electrónico;
- string com o mesmo significado de cadeia de caracteres;
- booleano com o mesmo significado de lógico;
- software com o mesmo significado de programa de computador;
- password com o mesmo significado de palavra-chave.

3. Quando se tratar de identificar trabalhos realizados no âmbito de projectos internacionais reconhecidos, que estejam suportados por documentos de âmbito público, serão utilizados os identificadores na língua original.

Exemplo: IMS Question & Test Interoperability Specification

4. Dado que este trabalho incidiu exclusivamente na concepção de uma camada de gestão das actividades de um curso, serão referidas todas as outras camadas dos LMS exactamente desse modo, como forma de simplificar o processo de descrição das diversas funcionalidades. Assim, far-se-á a simplificação correspondente a reduzir um LMS à nova camada de gestão que se propõe neste trabalho, e às “outras camadas de um LMS”.
5. Utilizar-se-á a palavra norma em substituição do termo *standard*. Todos os termos etimologicamente derivados de *standard* serão substituídos pelas correspondentes palavras portuguesas derivadas da palavra norma. Por exemplo, *standardization* será substituído por normalização.

6. Serão utilizadas aspas para realçar alguns termos utilizados ao longo do texto ou para estabelecer a diferença semântica entre essa utilização do termo e aquela que normalmente se lhe associaria.

Exemplo: Um aluno é uma entidade a quem se associa tipicamente a actividade de aprender, num processo de ensino/aprendizagem. Porém, quando no texto se utilizar a forma “aluno” ou “Aluno” isso significará que está a ser referido um utilizador da plataforma tecnológica (LMS) que, entre outras interacções com o sistema, também poderá executar actividades de aprendizagem.

7. Dado não ser relevante no contexto deste trabalho, e uma vez que mesmo a nível internacional não existe uma regra estabilizada quanto à sua definição e utilização, os termos norma, pré-norma, especificação, recomendação, etc, deverão ser entendidos como sinónimos, embora haja a consciência das diferenças semânticas entre os diversos termos.

8. Um dos projectos mais profundamente abordados nesta tese é o IMS Learning Design, cuja utilização como modelo de referência para o desenvolvimento de LMS implica a manipulação de alguns conceitos chave, um dos quais é o próprio *learning design*. Assim, quando se adoptar a forma *learning design* estar-se-á a fazer referência a esta entidade. Quando se utilizar a forma *Learning Design*, a intenção é referir a especificação ou norma IMS LD.

9. Serão utilizados os termos LMS e sistema com o mesmo significado, sendo que o contexto de utilização do termo sistema não deixará dúvidas quanto à sua utilização.

10. Os elementos de metadados do tipo data\_hora estão conformes a terceira edição da norma ISO 8601, a ISO 8601:2004, de 2004-12-13, com o formato:

YYYY-MM-DDThh:mm:ss

Nesta notação o carácter T separa a data da hora e todos os caracteres devem ser preenchidos.

Assim, a representação correspondente ao instante 1 hora e 5 minutos da manhã, do dia 5 de Agosto de 2006 deve ser a string:

2006-08-05T01:05:00.

**Esta página foi intencionalmente deixada em branco.**

# Capítulo 1

## Introdução

Desde há alguns anos que a UA (Universidade de Aveiro), quer directamente, quer através da UNAVE<sup>1</sup>, a unidade de interface da UA para a área da formação contínua, tem vindo a desenvolver a oferta de formação a distância suportada por ambientes Web.

Ao longo de todo esse tempo foi sendo constatado que o sucesso de uma mesma acção de formação era variável, mesmo com populações-alvo semelhantes.

A observação dessas diferenças nos resultados obtidos foi provocando o questionamento quanto à forma como os cursos eram disponibilizados e quanto ao modo como o correspondente acompanhamento era feito. Para algumas dessas questões levantaram-se algumas hipóteses de explicação. Um aspecto se evidenciou de entre o conjunto dos diversos potenciais motivos identificados como causadores da variabilidade do sucesso das acções de formação. Tratava-se da inconstância do processo de seguimento da participação dos diversos actores nas acções de formação.

Embora nenhum estudo específico e rigoroso tenha sido efectuado no sentido da demonstração da validade da hipótese colocada, pareceu, pela experiência acumulada, que se justificaria partir para o desenvolvimento de um trabalho de caracterização das actividades observáveis, e portanto passíveis de ser geridas, de um curso, que poderia permitir sistematizar um conjunto de funcionalidades de gestão de actividades associadas a um LMS (*Learning Management System*).

Foi então lançado o projecto “ODL Toolbox: Ferramentas para Apoio à Gestão de Sistemas de Ensino a Distância” (Ramos et al, 2003), financiado pela FCT (Fundação Para

---

<sup>1</sup> Associação para a Formação Profissional e Investigação da Universidade de Aveiro, fundada em 25 de Fevereiro de 1986, com o estatuto legal de “Entidade privada sem fins lucrativos”.

a Ciência e Tecnologia)<sup>2</sup>, no âmbito do qual se deram os primeiros passos no sentido da caracterização de tais actividades e no qual o autor deste documento se envolveu a partir de determinada altura.

O projecto ODL Toolbox, desenvolvido no Departamento de Comunicação e Arte em colaboração com o Departamento de Didáctica e Tecnologia Educativa, ambos pertencendo à UA, teve como parceiro a PT Inovação<sup>3</sup>, uma empresa do grupo PT (Portugal Telecom) e, por esse motivo, acabou por ser fortemente influenciado pelas preocupações de um âmbito não académico de formação/ensino.

De acordo com o modelo adoptado no âmbito do projecto ODL Toolbox (Ramos et al, 2003), é possível considerar as seguintes quatro fases no processo de criação, disponibilização e utilização de um curso suportado por LMS: “Concepção”, “Planeamento”, “Execução” e “Avaliação” (Ramos et al, 2001). Contudo, apenas foi considerada no âmbito desta tese, a fase “Execução”, visto ser aquela que mais directamente tem que ver com o envolvimento dos alunos.

Embora não sendo os únicos intervenientes no processo de ensino/aprendizagem, os alunos representam o elemento de maior peso, em termos de utilização de um LMS e de geração de informação associada à sua participação nos cursos suportados por tais sistemas.

A simplificação referida não constitui um caso isolado no tratamento desta problemática. Um dos projectos actuais mais importantes, desenvolvido no âmbito da adopção da norma IMS LD, o Coppercore (IMS, 2006), apenas considera actividades de aprendizagem e de suporte, adoptando portanto uma abordagem semelhante à que foi seleccionada neste trabalho. Dado que o projecto se encontra inserido numa *task force* importante a nível internacional, a similitude da opção tomada quanto ao foco nas actividades relacionadas com a fase “Execução” é reconfortante.

---

<sup>2</sup> <http://www.fct.mctes.pt/>

<sup>3</sup> <http://www.ptinovacao.pt/>

## 1.1 Contexto

Tentando reunir numa proposta global o seu trabalho de desenvolvimento de normas e especificações, a ADL (*Advanced Distributed Learning*)<sup>4</sup> organizou num conjunto de livros e no âmbito do projecto SCORM, informação destinada a servir de base à produção e empacotamento de conteúdos reutilizáveis, bem como à definição das condições normalizadas de incorporação desses conteúdos em cursos geridos por LMS interoperáveis, provenientes de diferentes fornecedores de software, e à definição das regras de sequenciamento e navegação nas actividades constituintes de um curso.

Tal conjunto de normas e especificações receberam o contributo de outros trabalhos de normalização levados a cabo pelos organismos e consórcios já referidos, principalmente do IMS<sup>5</sup>, do AICC (*Aviation Industry Computer Based Training Committee*)<sup>6</sup>, do IEEE LTSC (*Institute of Electrical and Electronic Engineers - Learning Technologies Standard Committee*)<sup>7</sup> e do ARIADNE (*Alliance of Remote Instructional Authoring and Distribution Networks for Europe*)<sup>8</sup>.

A figura 1 mostra o conjunto de livros a que se fez referência, realçando em cada caso a origem dos trabalhos de normalização que o projecto ADL SCORM integrou ou adoptou.

Uma observação atenta da figura 1 permitirá identificar a ausência de qualquer livro ou referência aos aspectos relacionados com a gestão operacional dos cursos na Web, isto é, à gestão das interacções dos diversos actores com as plataformas de *eLearning*, durante a execução das actividades que constituem os planos dos cursos.

Por esta razão, foi decidido incluir nos objectivos desta tese o estudo e formulação de uma proposta de integração das funcionalidades relacionadas com a camada de gestão operacional na norma SCORM.

---

<sup>4</sup> <http://www.adlnet.gov/index.cfm>

<sup>5</sup> <http://www.imsglobal.org/>

<sup>6</sup> <http://www.aicc.org/>

<sup>7</sup> <http://ieeeltsc.org/>

<sup>8</sup> <http://www.ariadne-eu.org>

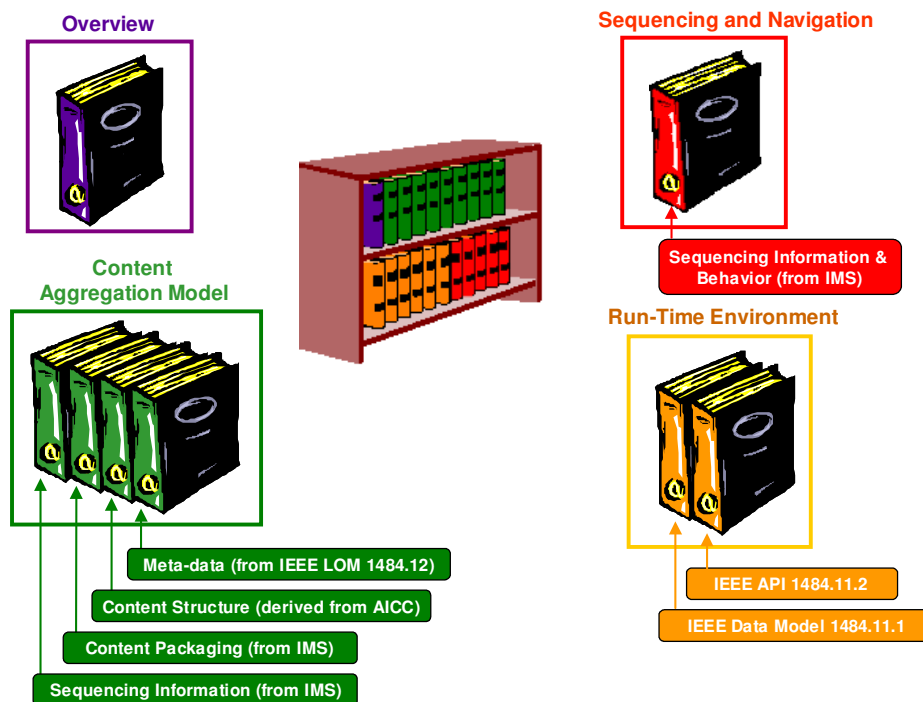


Figura 1 – Organização da norma SCORM com base em uma colecção de livros.  
Adaptado de (ADL, 2004a).

## 1.2 Gestão de actividades em cursos suportados por LMS

### 1.2.1 Antecedentes

Antes mesmo de se discutirem as questões de investigação e os objectivos a atingir com este trabalho, convirá convocar de novo o projecto ODL Toolbox, a partir do qual esta tese evoluiu.

Desse trabalho resultou uma primeira sistematização e descrição das actividades relevantes num processo de utilização de um curso suportado por plataformas Web, bem como a identificação de uma estrutura base e genérica de representação de uma actividade de ensino/aprendizagem fazendo parte de qualquer curso suportado por um LMS. Ao mesmo tempo foi feita a inventariação dos elementos de metadados respectivos.

A figura 2 apresenta tal estrutura, podendo encontrar-se na tabela 1 um excerto da tabela mais geral que pode ser encontrada no relatório final projecto ODL Toolbox

(Ramos et al, 2003), com a identificação e descrição de algumas das actividades identificadas como mais significativas.

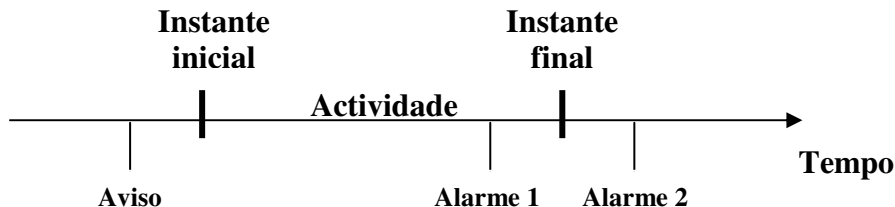


Figura 2 – Unidade atômica de gestão de actividades em tempo real.

Aquando da realização do projecto ODL Toolbox, o modelo contemplava a emissão de um aviso antes do início previsto para a execução de uma actividade, a emissão de um primeiro alarme antes de atingido o limite temporal para a execução da mesma e, finalmente, um segundo alarme imediatamente após ter sido atingido o instante final para execução da actividade.

O primeiro alarme seria emitido caso se detectasse que um actor não tinha ainda iniciado a actividade e o segundo alarme seria emitido em situações em que a actividade não estivesse concluída após ser atingido o limite temporal para a respectiva execução.

Não tinham sido, contudo, identificados a um nível abstracto, os diversos tipos de actividades e as acções a desencadear perante cada um deles.

Actividade	Parâmetros da actividade	Avisos	Alarmes	Procedimentos
Entrar no sistema (login)	Data 1 (início)	Entrar no curso	Não utilizou correctamente o login Não iniciou a sua participação no curso	Enviar mensagem ao suporte
	Data 2 (fim)			
	Forma de comunicação do aviso			
	Forma de comunicação do alarme			
	Antecedência do aviso			
	Tipo de interacção			
	Destinatário das mensagens			
Realizar a apresentação	Data 1 (início)	Realizar a apresentação	Não realizou a apresentação	Enviar mensagem ao suporte e tutor
	Data 2 (fim)			
	Forma de comunicação do aviso			
	Forma de comunicação do alarme			
	Antecedência do aviso			
Aceder aos conteúdos	Data 1 (início)	Aceder aos conteúdos desde Data 1	Não acedeu aos conteúdos	Enviar mensagem ao suporte e tutor
	Data 2 (fim)			
	Forma de comunicação do aviso			
	Forma de comunicação do alarme			
	Antecedência do aviso			
	Tipo de interacção			
	Destinatário das mensagens			

Tabela 1 – Excerto da tabela de actividades identificadas no projecto ODL Toolbox Obtida a partir do relatório final do projecto (Ramos et al, 2003).



A tabela 2 é também uma parte da informação mais abrangente relativa aos elementos de metadados que então foram identificados e que integram o relatório final do projecto ODL Toolbox que o leitor pode consultar em (Ramos et al, 2003).

O exemplo a seguir descreve o elemento de metadados *Activity*, correspondente à estrutura base representada na figura 2, tal como resultou do projecto ODL Toolbox.

Nr.	Name	Explanation	Multiplicity	Data Type
1.8	Activity	This category of management elements describes the activities that are characteristic of the course	1 or More (smallest permitted maximum: 10.000)	Container
1.8.1	ActivityID	The activity has a unique identification code	1 and only 1	ID
1.8.2	Designation	Activity name	1 and only 1	String (smallest permitted maximum: 100 characters)
1.8.3	Description	Activity characterization	0 or 1	String (smallest permitted maximum: 1000 characters)
1.8.4	Phase	The activity inserts on a course phase. Vocabulary: <ul style="list-style-type: none"> <li>• Design</li> <li>• Planning</li> <li>• Execution</li> <li>• Evaluation</li> </ul>	0 or 1	Vocabulary (Restricted)
1.8.5	BeginningDate	Date of the beginning of the activity	1 and only 1	Date
1.8.6	BeginningHour	Hour of the beginning of the activity	1 and only 1	Time
1.8.7	ClosingDate	Date of the closing of the activity	1 and only 1	Date
1.8.8	ClosingHour	Hour of the closing of the activity	1 and only 1	Time
1.8.9	Responsibility	The actors group who will perform the activity. The activity is characteristic of a group. Only one group. Vocabulary: <ul style="list-style-type: none"> <li>• Student</li> <li>• Professor</li> <li>• Support team</li> <li>• All</li> </ul>	1 and only 1	Vocabulary (Restricted)
1.8.10	Requirements	What is necessary for the activity to occur. Requirements aren't always available in some activities.	0 or 1	String (smallest permitted maximum: 1000 characters))
1.8.11	Interaction	Types of technologies used to the actor accomplish the activity. Some types of interaction aren't available in all activities.	0 or More	Vocabulary (Best Practice)

		Vocabulary: <ul style="list-style-type: none"> <li>• Email</li> <li>• Mailing lists</li> <li>• IRC</li> <li>• Videoconference</li> <li>• Access content</li> <li>• Upload file</li> <li>• Download file</li> <li>• Read</li> <li>• Simulation</li> <li>• Messaging</li> <li>• Newsgroups</li> <li>• Whiteboard</li> <li>• Groupware</li> <li>• Others</li> </ul>		
1.8.12	Procedures	Indication of actions (tasks) specific to accomplish during the activity on the part of the actors (other parameters of the activities)	0 or 1	String (smallest permitted maximum: 1000 characters)
1.8.13	Mandatory	Indication if the activity is mandatory or facultative to be executed by the actor(s): yes or no	0 or 1	Boolean

Tabela 2 – Excerto da tabela com a definição dos elementos de metadados identificadas no projecto ODL Toolbox - Obtida a partir do relatório final do projecto (Ramos et al, 2003).

### 1.2.2 Questões de investigação e objectivos

As questões de investigação que nortearam a realização do trabalho que deu origem ao presente documento são as seguintes:

- a) Que modelo conceptual deve servir de base à gestão operacional de um ambiente de ensino/aprendizagem, mediado por tecnologia?
- b) Que atributos informacionais devem ser contemplados na gestão operacional de um ambiente de ensino/aprendizagem?
- c) Como poderá ser efectuada a integração das funcionalidades de gestão operacional de ensino/aprendizagem na norma SCORM?

Para levar por diante o trabalho que permitiria responder a estas questões foi necessário identificar os objectivos a atingir no decurso do mesmo. Tais objectivos consistiram, no essencial, em:

- i. Definir uma camada de gestão em tempo real da execução das actividades constituintes de cursos suportados por LMS;
- ii. Identificar os elementos de metadados de gestão necessários e a forma de uma eventual integração desta nova camada com o que era a arquitectura proposta pelo projecto ADL SCORM.

### 1.3 Metodologia do trabalho

Definir antecipadamente uma metodologia de investigação é uma tarefa que se impõe para a realização de um trabalho deste tipo. O estabelecimento de metas bem definidas e formas de as atingir, ajuda a evitar desvios face aos objectivos traçados.

Numa primeira fase foram identificados o problema e o âmbito em que o mesmo ocorre.

Seguiu-se uma fase de revisão bibliográfica no sentido de retirar conclusões acerca da oportunidade e interesse do assunto em estudo, procurando-se identificar quais os trabalhos em curso na área, e quais, de entre eles, assumiam maior expressão internacional. Esta fase do trabalho, sempre importante num projecto como este, permitiu fazer a avaliação do que vulgarmente se designa por estado da arte.

Depois de identificados os trabalhos mais relevantes, e concluído que a área específica do trabalho a realizar não se encontrava coberta por qualquer projecto conhecido, foi tomada a decisão de adopção de um trabalho de referência, tendo no horizonte uma eventual futura integração dos resultados deste estudo.

As hipóteses que se colocaram diziam respeito aos projectos IMS e ADL SCORM, tendo-se optado pelo segundo, dado apresentar-se como um projecto agregador dos trabalhos já realizados por diversos organismos e consórcios, nomeadamente do próprio IMS.

Seguiu-se uma fase de caracterização exaustiva do problema, tendo sido utilizadas diversas técnicas de criação de modelos, capazes de permitir a captura do maior volume possível de informação relacionada com o tema.

Dado que, por razões relacionadas com o limite temporal existente para a realização da tese, não foi previsto o desenvolvimento de qualquer protótipo que permitisse realizar algum tipo de validação de resultados, componente fundamental para a continuidade de um

trabalho deste tipo, foi decidido que a validação das ideias subjacentes ao trabalho que deu origem a esta tese, seria efectuada com base na recolha da opinião de peritos na matéria. A validação efectuada até agora foi, assim, baseada nas observações emitidas pelos revisores das conferências internacionais às quais foi submetido um conjunto de artigos baseados no trabalho realizado.

## 1.4 Organização da tese

A tese encontra-se organizada segundo os seguintes 6 capítulos:

No Capítulo 1 é feita uma introdução ao assunto tratado na tese, são apresentadas as questões de investigação e a metodologia utilizada na realização do trabalho, terminando-se com uma apresentação da forma como o documento está organizado.

O Capítulo 2 trata do enquadramento do trabalho realizado, bem como das motivações para a escolha do tema, procurando-se dar uma panorâmica geral do “estado da arte” nesta área.

O Capítulo 3 é destinado a fazer um resumo dos principais trabalhos de normalização em curso, realçando-se aspectos históricos do aparecimento de tais projectos, respectiva evolução e objectivos que pretendem atingir. Sempre que possível e justificado, são também descritos os trabalhos já realizados ou em curso no âmbito dos diversos projectos.

No Capítulo 4 é feita uma abordagem às técnicas de criação de modelos de sistemas, realçando-se as diferentes perspectivas sob que podem ser representados. É feita também a apresentação de algumas das técnicas utilizadas para a captura e representação de conhecimento associado aos sistemas que estiveram na base da realização do trabalho que deu origem a esta tese.

O Capítulo 5 descreve, em detalhe, o trabalho realizado, desde a fase de análise e criação de modelos das interacções entre os utilizadores e os LMS, até à proposta de uma arquitectura que complementa as funcionalidades que se podem encontrar num LMS SCORM compatível actual. Tal proposta inclui as funcionalidades de gestão operacional dos cursos, tal como consideradas neste trabalho. O capítulo termina com uma proposta da forma como se poderá fazer a integração do trabalho proposto com o que já existe no âmbito do projecto SCORM.

Dado tratar-se do capítulo que descreve o trabalho realizado, a sua extensão é significativamente superior à dos restantes e justifica-se pelo facto de ser necessário um total esclarecimento do leitor quanto aos pressupostos que foram tidos em conta, quer na arquitectura da camada de gestão proposta, quer na sua eventual integração com um LMS desenvolvido de acordo com as recomendações SCORM.

O capítulo 6 é reservado à apresentação e discussão das conclusões e à identificação das principais limitações e dificuldades sentidas na realização do trabalho que levou à escrita da presente tese.

## Capítulo 2

### Enquadramento do trabalho

É sabido que a indústria de desenvolvimento de software há muito que procura a adopção de normas como forma de permitir que diferentes sistemas possam comunicar entre si. O caso do desenvolvimento de software de suporte ao ensino/aprendizagem e de criação de conteúdos de ensino/aprendizagem não é excepção, procurando-se que conteúdos desenvolvidos no âmbito de uma plataforma possam ser reutilizados noutra diferente. Tal prática permitirá uma grande economia de recursos, quer financeiros, quer humanos, já que componentes de ensino/aprendizagem normalizados poderão passar a ser disponibilizados no mercado, evitando custos de desenvolvimento de raiz, muitas vezes desnecessários, uma vez que o que se pretende pode já estar disponível.

Esta ideia é defendida, por exemplo, pelo consórcio AICC (AICC, 2006), ao afirmar pretender que a comunidade de treino na área da aviação obtenha o maior retorno possível por cada dólar aplicado no treino baseado em tecnologias, manifestando a opinião de que tal só será possível se forem promovidos os modelos normalizados de interoperabilidade.

Ainda de acordo com a mesma fonte, os vendedores de software poderão utilizar tais modelos no seio de um leque vasto de diferentes áreas de actividade. Este cenário permitirá efectuar a venda dos produtos a um mercado mais vasto e a um custo unitário menor.

## 2.1 Desenvolvimento de normas

O processo de geração de normas e especificações consiste num conjunto mais ou menos complexo de fases por que tem que passar o trabalho de um número vasto de equipas, até que os documentos gerados nesse contexto possam ser aceites internacionalmente como documentos de referência em determinada área.

A título de exemplo, apresenta-se a descrição do processo seguido no âmbito do consórcio IMS no sentido da geração e proposta de tais normas e especificações, no que ao problema da interoperabilidade de sistemas de ensino/aprendizagem a distância diz respeito.

Os elementos das equipas de trabalho IMS fazem o levantamento dos requisitos no âmbito de reuniões, encontros, *workshops* e outros eventos que ocorrem um pouco por todo o mundo, de forma a definir correctamente os aspectos críticos da interoperabilidade no mercado do ensino/aprendizagem.

Baseados nesses requisitos, desenvolvem especificações provisórias (*draft specifications*) fornecendo orientações para que o desenvolvimento de *software* seja feito de acordo com esses requisitos. Em todos os casos as especificações têm sido desenvolvidas atendendo às necessidades manifestadas internacionalmente, próprias de cada participante.

Uma vez terminado o trabalho interno das equipas IMS, e testada a interoperabilidade pretendida, normalmente envolvendo os membros do consórcio e outros participantes, a especificação provisória é aprovada formalmente pelo *IMS Technical Board* e é oficialmente passada a especificação definitiva à disposição do público em geral.

Paralelamente as especificações são submetidas a grupos de acreditação de normas, no sentido de contribuir para o reconhecimento e adopção, como parte integrante de um repositório mundial de normas técnicas sobre aprendizagem distribuída (*distributed learning*).

Apresentam-se seguidamente alguns dos principais organismos de acreditação internacionais com actividade na área do *eLearning*:

- ISO – International Standards Organization<sup>9</sup>;
- CEN – European Committee for Standardization<sup>10</sup>;
- IEEE LTSC<sup>11</sup>;
- IMS<sup>12</sup>;
- AICC<sup>13</sup>;
- ADL<sup>14</sup>;
- ANSI (American National Standards Institute)<sup>15</sup>.

## 2.2 Características dos LMS existentes no mercado

Em grande medida, os LMS que podem encontrar-se no mercado foram desenvolvidos sem terem sido consideradas as recomendações propostas pelos principais consórcios e organizações envolvidas no processo de normalização (IMS, ADL, IEEE LTSC, AICC, ARIADNE, etc), fazendo, por exemplo, com que conteúdos criados no âmbito de um LMS específico, não possam ser utilizados noutras plataformas (LMS).

Para além disso, exceptuando alguns casos particulares, a maior parte dos LMS apenas permite tarefas de autoria, disponibilização de conteúdos, criação de regras de sequenciamento e navegação pelas actividades constituintes dos cursos, e tratamento estatístico das participações dos vários actores que neles participam, deixando de lado aspectos relacionados com a gestão operacional das actividades de ensino/aprendizagem, tal como são propostas neste trabalho.

No entanto, o LMS FORMARE, desenvolvido e comercializado pela PT Inovação, inclui um conjunto de funcionalidades designadas genericamente por Alarmes, que se baseiam nos conceitos desenvolvidos no âmbito do já referido projecto ODL Toolbox.

A figura 3 representa o modelo genérico de um LMS, segundo a norma SCORM, sendo de notar o realce que o modelo dá aos diversos serviços que devem estar disponíveis num tal sistema (ADL, 2004a).

---

<sup>9</sup> <http://www.iso.org/iso/en/ISOOnline.frontpage>

<sup>10</sup> <http://www.cenorm.be/cenorm/index.htm>

<sup>11</sup> <http://ieeeltsc.org/>

<sup>12</sup> <http://www.imsglobal.org/metadata/index.html>

<sup>13</sup> <http://www.aicc.org/>

<sup>14</sup> <http://www.adlnet.gov/index.cfm>

<sup>15</sup> <http://www.ansi.org/>



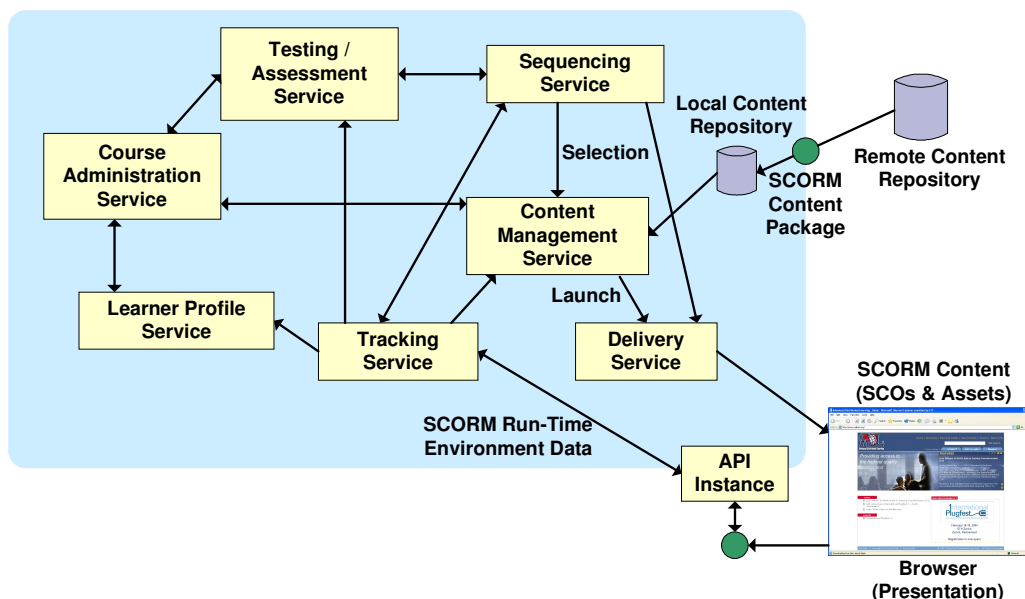


Figura 3 – Modelo genérico de um LMS. Adaptado de (ADL, 2004a)

De acordo com a norma SCORM, um LMS determina que actividades/conteúdos disponibilizar e quando o fazer, registando o progresso e o desempenho do aluno, à medida que este vai navegando nos conteúdos de aprendizagem (ADL, 2004a).

## 2.3 Motivações para a realização do trabalho

O que até agora se referiu neste documento permite identificar a existência de uma lacuna em termos de investigação e desenvolvimento na área do *eLearning*. Trata-se da gestão operacional das actividades de ensino/aprendizagem que podem ser encontradas no âmbito da utilização de LMS funcionando sobre a Web.

Um dos projectos mais ambiciosos nesta área, o IMS LD (*Instructional Management System – Learning Design*), fornece um referencial cada vez mais adoptado no desenvolvimento de plataformas de *eLearning* (RELOAD, CopperCore, eXe, etc) e, contudo, nenhuma ferramenta desenvolvida até ao momento sobre esta especificação, apresenta funcionalidades de gestão operacional de cursos na Web, apesar de no seu nível C, a especificação IMS LD já contemplar mecanismos de notificação, que poderiam ser utilizados com um objectivo semelhante ao que é proposto neste trabalho.

Esta utilização incompleta da especificação foi confirmada recentemente, em Julho de 2006, em contactos pessoais do autor com alguns dos principais elementos que se

encontram envolvidos em trabalhos relacionados com o projecto IMS LD, como foi o caso do Doutor Daniel Burgos<sup>16</sup>, do Doutor Rob Koper<sup>17</sup> e do Doutor Kinshuk<sup>18</sup>, na conferência ICALT2006 (*The 6th IEEE International Conference on Advanced Learning Technologies*).

No MOODLE (*Modular Object-Oriented Dynamic Learning Environment*), um LMS *open source* que se encontra em evolução para a integração das especificações *IMS Learning Design* (Berggren, 2006), e que já permite a inclusão de conteúdos compatíveis com a norma SCORM 1.3 (*Shareable Content Object Reference Model 1.3*), também referida como SCORM 2004, já existe a possibilidade de utilização de notificações. Contudo, este mecanismo apenas é utilizado para informar o professor de que um aluno submeteu ou actualizou determinado trabalho.

Para além destes, apenas foi encontrado, muito recentemente, um trabalho fazendo referência a um mecanismo de notificações associado à execução de actividades (Wu, 2006), embora não tendo na base da sua utilização exactamente as mesmas preocupações que estiveram na origem da presente tese de doutoramento. No referido trabalho apenas foram considerados os alunos como actores relevantes do processo, e não houve nenhuma preocupação de integração do trabalho realizado, em qualquer projecto de normalização em curso. O mecanismo de notificações aí apresentado apenas se destina a informar os alunos sobre que actividades estes devem executar no âmbito do seu processo de aprendizagem, e sobre o seu estado de evolução nesse processo.

Cinco anos passados sobre o surgimento da ideia de realizar um trabalho sobre este tema, e com tantos projectos de normalização ainda a decorrer (ADL, 2004a), (ADL, 2004b), (ADL, 2004c), (ADL, 2004d), (IMS, 2001), (IMS, 2003a), (IMS, 2003b), (IMS, 2003c), (IMS, 2003d), é possível constatar que a camada de gestão operacional das actividades que constituem um curso, continua a não ser contemplada nesses trabalhos.

O próprio projecto que serviu de referencial à realização do presente trabalho – o ADL SCORM – talvez a norma mais adoptada internacionalmente para a construção de plataformas de *eLearning* interoperáveis e para o empacotamento de conteúdos reutilizáveis, não faz qualquer referência aos aspectos de gestão operacional das actividades de ensino/aprendizagem mencionados nesta tese.

---

<sup>16</sup> <http://www.open.ou.nl/dbu>

<sup>17</sup> <http://www.iel.nl/rkp.htm>

<sup>18</sup> <http://scis.athabascau.ca/scis/staff>

Este facto poderia levar a pensar que o tema não é visto como interessante por parte deste e de outros projectos de normalização. Contudo, o que se passou com o projecto IMS LD aponta noutra direcção. Provavelmente, apenas questões de agenda ou de oportunidade terão feito com que a questão fosse deixada para futuras evoluções. De facto, após um início em que apenas aspectos relacionados com o vocabulário nuclear de suporte à vasta diversidade pedagógica existente (nível A) e com a inclusão de “Propriedades” e “Condições” capazes de permitir personalizar e obter sequências de actividades e interacções mais elaboradas (nível B), o projecto passou a contemplar, no seu nível C, as notificações.

A inclusão desta funcionalidade só aconteceu bastante tempo depois de ter surgido a ideia de desenvolver o trabalho que deu origem à presente tese e não será descabido de sentido, pensar que o mesmo se virá a passar com o projecto ADL SCORM, seja por desenvolvimento próprio, seja por adopção/incorporação do que entretanto existir ao nível de outros projectos de normalização.

## Capítulo 3

### Principais projectos de normalização

Os sistemas CBT (*Computer Based Training*) têm sido desenvolvidos por diversas entidades (Keystar<sup>19</sup>, CBT Direct<sup>20</sup>, CBT Planet<sup>21</sup>, CBT Systems<sup>22</sup>, CBT Solutions<sup>23</sup>), utilizando um variado conjunto de ferramentas ou sistemas de autor. Muitas das “aulas” concebidas para sistemas de formação suportados por CBT poderão complementar-se e integrar-se de forma satisfatória com outras “aulas” criadas com diferentes ferramentas, por diferentes pessoas. Existe uma grande necessidade de permitir que essa complementaridade se afirme em cursos formados à custa de partes provenientes de origens diversas. Contudo, isso só poderá ser feito se existir um conjunto de funções CMI (*Computer Managed Instruction*) normalizado e o correspondente conjunto de funções CBT (Goh, 2005), (AICC, 2004).

Adicionalmente, existem fortes razões para que um curso possa ser criado numa determinada plataforma e possa ser migrado para outra diferente. Essas razões prendem-se com:

- a) Poupança na aquisição, pois bastará adquirir um sistema CMI (uma plataforma) em vez de um sistema específico para cada curso a ser disponibilizado;
- b) Diminuição de custos de administração dos sistemas, já que apenas haverá que administrar um.

---

<sup>19</sup> <http://www.keystar.co.uk/training-online/microsoft-cbt-training.php>

<sup>20</sup> <http://www.cbtdirect.com/>

<sup>21</sup> <http://www.cbtplanet.com/>

<sup>22</sup> <http://education-place.info/1/365/Cbt%20Software/index.html>

<sup>23</sup> <http://www.cbt-solutions.co.uk/coursewriter.htm>

Por estes motivos se impõe a normalização da descrição e apresentação de conteúdos e do sequenciamento das “aulas” dentro de um curso.

Finalmente, haverá muito a ganhar com a análise dos dados acumulados, relativos ao desempenho de um aluno durante o seu envolvimento numa “aula”. De tal análise poderá resultar a redefinição das tarefas a executar, a intervenção dos professores ou tutores, ou mesmo a reconfiguração automática do curso para um dado aluno ou conjunto de alunos.

Um formato normalizado de descrição do desempenho de um aluno pode permitir a análise de diferentes “aulas”, criadas por diferentes autores, utilizando diferentes sistemas.

## 3.1 O projecto IEEE LTSC

### 3.1.1 Introdução

O IEEE LTSC apareceu, tal como o IMS Project, em 1997, para o desenvolvimento de normas para as tecnologias de informação utilizadas no ensino/aprendizagem, no treino e na educação, e é mantido pelo *IEEE Computer Society Standards Activity Board* para desenvolver padrões técnicos acreditados, práticas recomendadas, e recomendações para as tecnologias de aprendizagem.

As tecnologias de aprendizagem incluem os componentes de *software*, as ferramentas, as tecnologias, e os métodos de projecto que facilitem o desenvolvimento, a distribuição, a manutenção, e a interoperação de componentes e sistemas de educação e treino suportados por computador.

### 3.1.2 Objectivos

De acordo com a informação disponível (IEEE LTSC, 2006), os objectivos do comité LTSC do IEEE são os seguintes:

- Permitir que os alunos ou os professores possam procurar, avaliar, adquirir e utilizar objectos de aprendizagem;
- Permitir a partilha e a troca de objectos de aprendizagem entre quaisquer sistemas de aprendizagem suportados por tecnologias;

- Possibilitar o desenvolvimento de objectos de aprendizagem com base em unidades que possam ser combinadas ou decompostas de uma forma coerente;
- Permitir que agentes computacionais componham de uma forma automática e dinâmica, aulas personalizadas de acordo com as características de um dado aluno;
- Complementar o trabalho directamente relacionado com normas que estejam focadas no tornar possível que múltiplos objectos de aprendizagem (*Learning Objects*) possam coexistir e ser utilizados conjuntamente num ambiente aberto e distribuído de aprendizagem;
- Permitir, quando desejado, o reconhecimento e a documentação de objectivos de aprendizagem e desempenhos, já existentes ou novos, associados aos objectos de aprendizagem;
- Permitir uma forte e crescente economia ao nível dos objectos de aprendizagem que suportam as diversas formas de distribuição: sem fins lucrativos, e com fins lucrativos;
- Permitir que organizações dedicadas à educação, ao treino e à aprendizagem, seja na área governamental, seja em áreas públicas ou privadas, possam expressar conteúdos educacionais e desempenhos normalizados, num formato normalizado, que seja independente do conteúdo em si mesmo;
- Dotar os investigadores de normas que suportem a recolha e a partilha de informação comparável relacionada com a aplicabilidade e eficácia dos objectos de aprendizagem;
- Definir uma norma que seja simples e extensível a múltiplos domínios e jurisdições, de forma a tornar mais fácil a sua adopção e aplicação;
- Suportar a segurança e autenticação necessárias à distribuição e utilização dos objectos de aprendizagem.

### 3.1.3 Normas, especificações ou recomendações geradas

Desde a criação do comité LTSC do IEEE, diversos grupos de trabalho foram criados e, enquanto alguns deles geraram normas ou padrões, outros estão ainda activos na busca de uma versão final de documentos normalizadores, enquanto que outros desapareceram por inactividade continuada.

De acordo com a informação disponível no sítio oficial do IEEE<sup>24</sup>, os grupos de trabalho (*WG – Work Group*) que se encontram activos são os seguintes:

#### **WG4: Digital Rights Expression Languages (DREL)**

Este grupo de trabalho encontra-se em fase de levantamento das características que uma DREL (*Digital Rights Expression Language*) normalizada deve possuir, de forma a identificar e recomendar um conjunto de práticas destinadas à definição/utilização de linguagens de descrição de direitos digitais, adequadas às tecnologias do *eLearning*.

A protecção dos direitos sobre conteúdos digitais tornou-se um tema com muita visibilidade para o público em geral, que tomou contacto com esse problema colocado ao nível dos direitos associados a manifestações da tecnologia. É o caso dos filmes em DVD (*Digital Versatile Disk*) ou dos leitores de música em formato MP3 (*MPEG Audio Layer-3*). Isto não é menos verdade no mundo da educação em que conteúdos de ensino e documentos de pesquisa digitalizados estão a ser manipulados no âmbito de bibliotecas, portais Web e ambientes virtuais de aprendizagem (*VLE – Virtual Learning Environment*).

Esta mudança está a fazer emergir novos problemas relacionados com a protecção dos direitos de autor, e com a necessidade de compreender os problemas tecnológicos envolvidos.

A tecnologia, na forma de sistemas de gestão de direitos digitais (*DRM - Digital Rights Management*) que reforçam as políticas de protecção dos direitos, tem um papel da maior importância a desempenhar. Contudo surge a dúvida sobre como expressar esses direitos. Para os humanos os direitos estão documentados em asserções formais, codificadas em leis e expressas em contratos, termos e condições. Contudo, no sentido da manipulação dos direitos por sistemas computadores, estes têm que ser expressos formalmente, de uma forma que permita que os computadores consigam lê-los.

Para um maior aprofundamento do tema sugere-se a leitura do relatório da autoria de Chris Barlas (Chris Barlas, 2006).

O mesmo tipo de problemas é discutido em (Santos, 2002) e uma proposta de solução para os mesmos é também apresentada pelo autor nesse documento.

---

<sup>24</sup> <http://www.ieee.org/portal/site>

## **WG11: Standard for Information Technology - Learning Systems - Computer Managed Instruction (CMI)**

O grupo de trabalho WG11 desenvolve a sua actividade no sentido da obtenção de normas relacionadas com:

- A descrição do conteúdo de um curso;
- A organização e sequenciamento das “aulas” individuais (sessões ou unidades atribuíveis);
- A comunicação de informação entre o software de gestão de um conjunto de “aulas” (isto é, um sistema CMI) e as “aulas” propriamente ditas;
- A descrição dos objectivos de um curso e o seu relacionamento com “aulas” ou conjuntos de “aulas”;
- A apresentação da informação relativa ao desempenho dos alunos e o relacionamento desse desempenho com os objectivos do curso.

Não estão cobertos pelos trabalhos realizados, aspectos como:

- Arquitectura ou definição de *curricula*;
- Conteúdos das “aulas”;
- Comportamento ou estrutura de uma “aula”;
- Análise de dados relativos ao desempenho dos alunos.

As normas desenvolvidas no âmbito deste grupo de trabalho têm como objectivos:

- Permitir que diferentes “aulas” possam ser utilizadas em diferentes sistemas CMI;
- Permitir o intercâmbio de cursos entre diferentes sistemas CMI (permitir interoperabilidade);
- Permitir a edição e alteração de cursos por parte de entidades diferentes daquelas que os criaram, com utilização de diferentes ferramentas CMI;
- Permitir a análise fácil dos dados relativos ao desempenho dos alunos no âmbito do seu envolvimento em diversas sessões.



O grupo de trabalho WG11 envolveu-se activamente na geração e disponibilização das seguintes normas:

#### **1484.11.1: Data Model for Content to Learning Management System Communication**

**Título:** *IEEE Standard for Learning Object Metadata*

**Âmbito:** Esta norma é composta por diversas partes e especifica metadados de objectos de aprendizagem (*Learning Object Metadata*). A norma descreve formalmente um esquema conceptual de dados que define a estrutura de uma instância de metadados para um objecto de aprendizagem. No contexto da mesma, um objecto de aprendizagem é definido como qualquer entidade, digital ou não, que pode ser utilizada para promover aprendizagem, educação ou treino.

#### **1484.11.2: ECMAScript API for Content to Runtime Services Communication**

**Título:** *Standard for Learning Technology - ECMAScript Application Programming Interface for Content to Runtime Services Communication*

**Âmbito:** Norma que descreve uma API (*Application-Programming Interface*) escrita em ECMAScript (linguagem de *script* com origem na *ECMA - European Computer Manufacturers Association*<sup>25</sup>, constituindo a base da actual JavaScript), para suporte à comunicação entre os conteúdos e os serviços dos LMS durante a execução (*runtime*). A norma baseia-se numa outra API definida no documento “*CMI Guidelines For Interoperability*”, versão 3.4, gerada no seio do AICC. Nela são definidos serviços comuns da API, serviços esses que possibilitam a comunicação de informação entre conteúdos relacionados com a aprendizagem e um serviço do LMS (*RTS - Runtime Service*) utilizado para suportar a gestão da aprendizagem. Esta norma não cobre aspectos relacionados com as estruturas de dados utilizadas na transmissão da informação, nem questões

---

<sup>25</sup> <http://www.ecma-international.org/>

de segurança de dados, nem o processo comunicacional entre um RTS e os sistemas de gestão respectivos.

**Objectivos:** Existe um reconhecimento alargado de que a API ECMAScript para suporte à comunicação entre conteúdos e serviços dos LMS durante a execução, definidos no documento “*CMI Guidelines For Interoperability*”, versão 3.4, tem uma grande amplitude de aplicação em sistemas utilizados na gestão da aprendizagem. O objectivo desta norma consiste na construção de consensos, na resolução de ambiguidades, e na correcção de defeitos que a API apresenta, no processo de supervisão das comunicações referidas.

#### **WG12: Standard for Information Technology - Education and Training Systems - Learning Objects and Metadata**

O grupo de trabalho 12 realiza trabalho destinado à obtenção de normas capazes de orientar a especificação da sintaxe e da semântica dos metadados dos objectos de aprendizagem (*LOM - Learning Object Metadata*), definidos como os atributos necessários à completa descrição de objectos de aprendizagem (*Learning Objects*). Os objectos de aprendizagem são definidos como qualquer entidade, digital ou não, que podem ser utilizados, reutilizados ou referenciados durante a aprendizagem suportada por tecnologias de informação e comunicação (IEEE, 2002).

Apresentam-se de seguida alguns casos de tecnologias de suporte à aprendizagem, tal como são consideradas pelo comité LTSC do IEEE:

- Sistemas de treino assistido por computador;
- Ambientes interactivos de aprendizagem;
- Sistemas inteligentes de ensino apoiados por computadores;
- Sistemas de aprendizagem à distância;
- Ambientes de aprendizagem colaborativa.

O comité LTSC do IEEE (IEEE LTSC, 2006) apresenta como exemplos de objectos de aprendizagem, os seguintes:

- Conteúdos multimédia;
- Objectivos de aprendizagem;
- Programas e ferramentas de ensino;
- Pessoas, organizações ou eventos referenciados durante sessões de aprendizagem suportada por tecnologias.

As normas para a identificação e descrição dos metadados de objectos de aprendizagem, incidem num conjunto mínimo de atributos, necessário à manipulação, localização e avaliação desses objectos de aprendizagem e têm, ainda, a capacidade de localmente estender os atributos básicos e os tipos de entidades, sendo que tais atributos podem ser de utilização obrigatória ou opcional.

Alguns dos atributos mais relevantes dos objectos de aprendizagem que devem ser descritos, incluem os tipos dos objectos, os autores, os donos, as condições em que podem ser disponibilizados e os formatos válidos. Em situações em que tal seja aplicável, os metadados dos objectos de gestão podem também incluir atributos pedagógicos tais como o tipo de interacção ou ensino, nível de classificação, pré-requisitos ou tipo de supervisão.

É possível, para qualquer objecto de aprendizagem, definir mais do que um conjunto de metadados de objectos de aprendizagem. A norma não prevê qualquer tipo de recomendação relativa à forma como essas definições são feitas. Espera-se que as normas evoluam para uma uniformização que viabilize uma possível integração ou referência a outras normas abertas disponíveis e a outros trabalhos, em outras áreas relacionadas (IEEE LTSC, 2006).

#### **1484.12.2: Standard for ISO/IEC 11404 binding for Learning Object Metadata data model**

**Título:** *Standard for ISO/IEC 11404 binding for Learning Object Metadata data model*

**Âmbito:** Esta norma especifica um enquadramento conjunto ISO (*International Standards Organization*) e IEC (*International Electrotechnical Commission*) ISSO/IEC11404:1996 (*Language Independent Datatypes*) do modelo de dados IEEE 1484.12.1. Qualquer implementação que esteja em conformidade com a norma 1484.12.2 deve estar também conforme a norma 1484.12.1.

**Objectivos:** O objectivo desta norma consiste em fornecer uma semântica rigorosa para o modelo de dados, de acordo com a notação 11404<sup>26</sup>. Esta notação pode ser útil para o estabelecimento de regras relativas ao desenvolvimento de linguagens de programação e de outros sistemas.

### **1484.12.3: Standard for Learning Technology-Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata**

**Título:** *Standard for XML binding for Learning Object Metadata data model.*

**Âmbito:** Esta norma especifica o formato XML para o modelo de dados IEEE 1484.12.1<sup>27</sup>. Qualquer implementação que esteja em conformidade com a norma 1484.12.3, também deve estar conforme a norma 1484.12.1.

**Objectivos:** O objectivo desta norma consiste no fornecimento de um formato XML que permita o intercâmbio de instâncias LOM entre diferentes sistemas que implementem o modelo de dados 1484.12.1.

### **1484.12.4: Standard for Resource Description Framework (RDF) binding for Learning Object Metadata data model**

**Título:** *Standard for Resource Description Framework (RDF) binding for Learning Object Metadata data model.*

**Âmbito:** Esta norma especifica o formato W3C (*World Wide Web Consortium*)<sup>28</sup> RDF do modelo de dados IEEE 1484.12.1. Qualquer implementação que esteja em conformidade com a norma 1484.12.4, deve estar também conforme a norma 1484.12.1.

---

<sup>26</sup> [http://ltsc.ieee.org/wg12/files/IEEE\\_1484\\_12\\_02\\_D01\\_LOM\\_11404\\_binding.doc](http://ltsc.ieee.org/wg12/files/IEEE_1484_12_02_D01_LOM_11404_binding.doc)

<sup>27</sup> [http://ltsc.ieee.org/wg12/files/LOM\\_1484\\_12\\_1\\_v1\\_Final\\_Draft.pdf](http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf)

<sup>28</sup> <http://www.w3.org/>

**Objectivos:** O objectivo desta norma é o fornecimento de um formato RDF que permita o intercâmbio de instâncias de metadados de objectos de aprendizagem (*LOM – Learning Object Metadata*) entre diferentes sistemas que estejam em conformidade com o modelo de dados definido pelo grupo de trabalho 1484.12.1.

## **WG20: Competency data standards**

Esta norma deverá especificar os elementos obrigatórios e os elementos opcionais que constituem uma Definição de Competências, tal como é utilizada num LMS ou referenciada num Perfil de Competências.

O objectivo da norma é a definição de um modelo de Definição de Competências, universalmente aceite, que permita a criação, a troca e a reutilização de Definições de Competências em aplicações tais como Learning Management Systems, Análise de Insuficiência de Competências ou Capacidades, Perfis de Competência de Alunos ou outras entidades, etc.

A norma é necessária devido ao facto de presentemente existirem muitas definições de termos tais como “Objectivos de Aprendizagem”, “Competências”, “Capacidades”, etc, não se verificando um consenso quanto à forma como essas definições devem ser utilizadas por forma a permitir a definição de modelos de dados reutilizáveis.

## **3.2 O projecto IMS Global Learning Consortium**

### **3.2.1 Introdução**

Ao mesmo tempo que o IMS se iniciava com objectivos centrados no ensino superior, as especificações publicadas à data, e aquelas que haveriam de aparecer mais tarde, cobriam necessidades dum largo espectro dos contextos de aprendizagem, incluindo as escolas K-12 (ensino de nível secundário) e o treino/ensino privado e público.

O nome formal para o IMS é *IMS Global Learning Consortium, Inc.* O nome original, quando o projecto IMS se iniciou em 1997, era *Instructional Management Systems (IMS)* e tinha nascido no seio da *National Learning Infrastructure Initiative (NLII)* do

*EDUCAUSE Learning Initiative (ELI)*<sup>29</sup>, uma associação sem fins lucrativos, criada em 1994, cuja missão é fazer evoluir o ensino superior, promovendo a utilização inteligente das tecnologias da informação<sup>30</sup>. Nessa altura (1994) o EDUCAUSE era designado por Educom.

O consórcio (IMS) é uma organização mundial sem fins lucrativos incluindo mais de 50 membros activamente participativos. Estes são originários das mais diversas áreas da comunidade global de *eLearning* e incluem produtores de hardware e software, instituições de ensino e educação, editores, agências governamentais, integradores de sistemas, fornecedores de conteúdos multimédia e outros consórcios (IMS, 2001b).

Diversas especificações IMS tornaram-se normas efectivas para o fornecimento de produtos e serviços de aprendizagem a nível mundial, podendo encontrar-se imensa informação relacionada com essas normas no sítio “[www.imsglobal.org](http://www.imsglobal.org)”.

O âmbito das especificações IMS, normalmente orientadas para a aprendizagem distribuída (*distributed learning*) incluem tanto orientações para contextos *on-line* (síncronos) como para contextos *off-line* (assíncronos). Isto significa que os contextos de aprendizagem beneficiadores das especificações IMS incluem ambientes específicos da Internet (como é o caso dos sistemas de gestão de cursos suportados pela Web), da mesma forma que incluem soluções de aprendizagem envolvendo recursos electrónicos *off-line* (como é o caso de sistemas de ensino/aprendizagem que implicam a utilização de CD-ROM (*Compact Disk - Read Only Memory*) ou DVD-ROM (*Digital Versatile Disk - Read Only Memory*)).

Por exemplo, a especificação *IMS Learning Resources Meta-data Specification*<sup>31</sup>, beneficia o aluno que se encontra à procura de informação, uma vez que a associação de metadados às estruturas de armazenamento de informação, permite a utilização de ferramentas de pesquisa mais eficazes, quer ele se encontre a pesquisar recursos suportados pela Web, quer se encontre a fazer a pesquisa no seu computador, em casa, sobre um CD-ROM ou um DVD-ROM.

---

<sup>29</sup> <http://www.educause.edu/>

<sup>30</sup> <http://www.educause.edu/about>

<sup>31</sup> [www.imsglobal.org/metadata](http://www.imsglobal.org/metadata)

### 3.2.2 Objectivos

De acordo com a informação disponível no sítio oficial do consórcio, os objectivos do *IMS Global Learning Consortium* são os seguintes:

- a) Gerar ou contribuir para a criação de normas orientadas à satisfação das necessidades do mundo real no que toca à reutilização e à interoperabilidade de sistemas de ensino/aprendizagem, que resultem de uma discussão alargada entre os membros do consórcio, com interesses em áreas de negócio diversificadas;
- b) Desenvolver e promover a adopção de especificações técnicas abertas, orientadas à interoperação de tecnologias de ensino/aprendizagem.

### 3.2.3 Normas, especificações ou recomendações geradas

No seio do *IMS Global Learning Consortium* existe um vasto conjunto de trabalhos de normalização a que se fará referência de seguida. Contudo, dado que nem todos esses trabalhos estão directamente relacionados com a problemática tratada no âmbito desta tese, apenas serão referidos aqueles que foram considerados mais relevantes no contexto do presente documento.

#### **IMS AccessForAll Meta-data Specification**

A especificação de metadados *AccessForAll* trata de tornar possível a identificação de recursos que coincidem com as necessidades ou preferências dos utilizadores. Essas preferências ou necessidades podem ser declaradas utilizando a especificação IMS LIP (*Learner Information Package*), que será apresentada mais adiante nesta secção e incluem a necessidade ou preferência pela apresentação de recursos alternativos, métodos alternativos de controlo desses recursos, alternativas aos próprios recursos e melhorias ou suporte solicitados pelos utilizadores.

A especificação fornece uma linguagem comum para identificação e descrição dos recursos primários ou recursos considerados por defeito, e alternativas equivalentes para esses recursos.

Este trabalho representa uma colaboração aberta entre membros das comunidades IMS, DCMI (*Dublin Core Metadata Initiative*)<sup>32</sup>, IEEE, CEN (*European Committee for Standardization*)-ISSS (*Information Society Standardization System*)<sup>33</sup>, Eduspecs (*Industry Canada's EduSpecs Initiative*)<sup>34</sup> e outras. A especificação de metadados *AccessForAll* é uma abordagem baseada numa proposta unificada destinada a estabelecer a ligação entre as necessidades e preferências dos utilizadores e os recursos que permitem satisfazer essas necessidades, utilizando as diversas participações das especificações provenientes de diversas comunidades.

### **IMS Reusable Definition of Competency or Educational Objective Specification**

A especificação RDCEO (*Reusable Definition of Competency or Educational Objective*) fornece um meio para criar definições comuns acerca do que sejam competências que aparecem como parte de um plano de aprendizagem ou de uma carreira profissional, como pré-requisitos de aprendizagem ou como resultados da aprendizagem. O modelo de informação nesta norma pode ser utilizado para promover a troca destas definições entre diferentes sistemas de aprendizagem, sistemas de recursos humanos, conteúdos de aprendizagem, competências ou capacidades. A especificação fornece referências únicas para descrições de competências ou objectivos e para a inclusão em outros modelos de informação<sup>35</sup>.

Esta especificação é de tal forma importante que o IEEE/LTSC solicitou e recebeu autorização para a utilizar como base para a definição de uma norma de competência IEEE. A obtenção de tal norma implicará a conversão do formato RDCEO para o formato IEEE e o processo de conversão deverá ser realizado de tal forma que o resultado final possa vir a ser considerado uma norma.

---

<sup>32</sup>O DCMI é uma organização aberta preocupada com o desenvolvimento de normas de metadados em linha, interoperáveis e capazes de suportar um vasto conjunto de objectivos e modelos de negócio.

<sup>33</sup> O CEN/ISSS foi criado em meados de 1997 pelo CEN para acolher as suas actividades relacionadas com as TIC. Mais informação sobre esta iniciativa pode ser encontrada no sítio: [http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/iss/about\\_iss/index.asp](http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/iss/about_iss/index.asp).

<sup>34</sup> A Eduspecs representa a posição do governo federal Canadiano a nível internacional, no que toca à estratégia de inovação, e tem como missão promover e facilitar a adopção de normas e especificações interoperáveis, relativas ao fenómeno do eLearning.

<sup>35</sup> Mais informação poderá ser encontrada no sítio: <http://www.imsglobal.org/competencies/index.html>.



O documento relativo à especificação RDCEO explica a forma como ambos os projectos irão colaborar e fornece informação acerca da forma como outros potenciais parceiros poderão juntar-se para a obtenção desta norma.

## **IMS Content Packaging Specification**

A especificação *IMS Content Packaging Specification* fornece as funcionalidades necessárias à descrição e ao empacotamento de materiais de aprendizagem, tais como cursos ou conjuntos de cursos, em pacotes interoperáveis, e documenta a descrição, a estrutura e a localização de materiais de aprendizagem em linha e a definição de alguns tipos de conteúdos especiais.

Os principais interessados nesta especificação são os produtores de conteúdos, as empresas que desenvolvem e comercializam LMS e fornecedores de serviços do mercado do ensino/aprendizagem a distância.

Os materiais de estudo descritos e empacotados, utilizando o formato XML (*Extensible Markup Language*) do *IMS Content Packaging*, devem ser interoperáveis no seio de quaisquer ferramentas que suportem a especificação.

Os produtores de conteúdos deverão ter presente que o resultado do seu trabalho deve poder ser utilizado em qualquer sistema compatível com a especificação. Esta prática tornará a sua actividade mais rentável, já que protege o investimento feito na produção dos mesmos. Essa protecção deriva do facto de os conteúdos poderem ser produzidos uma vez e fornecidos para clientes possuidores de diferentes plataformas, aumentando o número de potenciais interessados.

A especificação, na sua versão final 1.1.2, ficou disponível em 2001 e é composta por informação que se encontra distribuída pelos seguintes três documentos:

- *IMS Content Packaging Information Model*
- *IMS Content Packaging XML Binding Specification*
- *IMS Content Packaging Best Practices and Implementation Guide*

## IMS Digital Repositories Specification

A proposta da especificação final v1.0 para os repositórios digitais IMS (*The IMS Digital Repositories v1.0 Final Specification*), disponibilizada em 30/01/2003, fornece um conjunto de recomendações relativas à interoperação das funcionalidades mais comuns sobre repositórios. Essas recomendações devem ser implementáveis através de serviços que permitam a apresentação de uma interface comum.

A um nível mais genérico, esta especificação define repositórios digitais como sendo qualquer colecção de recursos que estejam acessíveis via rede sem necessidade de qualquer conhecimento suplementar e prévio sobre a estrutura de tal colecção.

Os repositórios podem armazenar os recursos actuais ou os metadados que descrevem esses recursos. Os recursos e os metadados que os descrevem, não precisam de estar num mesmo repositório.

Esta especificação utiliza esquemas anteriormente definidos, por exemplo o *IMS MetaData and Content Packaging*, em vez de um novo esquema.

## IMS Enterprise Services Specification

A especificação *Enterprise Services Specification* consiste na definição da forma como os sistemas gerem a troca de informação que caracteriza pessoas, grupos e participantes, num contexto de aprendizagem, e foi aprovada pelo *IMS Technical Board*, na sua versão 1, em Julho de 2004. A especificação é construída seguindo as seguintes recomendações documentadas no documento *IMS Abstract Framework*:

- **Interoperability** – A especificação está centrada na troca de informação entre “sistemas empresariais”. Nada está previsto na especificação, relativamente à forma como os dados são geridos dentro de tais sistemas;
- **Service-oriented** - A especificação define a troca de informação em termos dos serviços a ser fornecidos no âmbito da colaboração dos sistemas. Esta colaboração é materializada em *Person Management Services*, *Group Management Services* e *Membership Management Services*;
- **Component-based** – O conjunto de serviços será fornecido de tal forma que possa dar origem a diversas combinações. Os serviços *Person Management Services*,

*Group Management Services* e *Membership Management Services* podem ser combinados de forma a fornecer outros serviços;

- **Layering** – O serviço *Enterprise Service* e os seus serviços constituintes (*Person, Groups e Membership*) são parte integrante da camada *Application Services*;
- **Behaviors and Data Models** - Os *Enterprise Services* são definidos em termos dos seus comportamentos e dos seus modelos de dados. Os comportamentos provocam alterações no estado do modelo de dados e o estado do modelo de dados deve ser apenas alterado como resultado de um comportamento bem definido;
- **Multiple Bindings** – O modelo de informação associado aos *Enterprise Services* deve ser definido através da utilização das técnicas da UML (*Unified Modeling Language*).
- **Adoption** - Os *Enterprise Services* são baseados na especificação original do modelo de dados *Enterprise*.

### **IMS ePortfolio Specification**

A especificação *IMS ePortfolio* foi criada com o objectivo de tornar os ePortfolios interoperáveis entre diversos sistemas e instituições. A especificação:

- Suporta os avanços verificados na aprendizagem ao longo da vida;
- Possibilita a transferência fácil de portfolios, da escola para os postos de trabalho;
- Permite que as instituições e os formadores possam acompanhar mais efectivamente a evolução das competências dos formandos;
- Melhora as experiências de aprendizagem e promove o desenvolvimento dos empregados.

### **IMS Learner Information Package Specification**

A especificação *IMS LIP (Learner Information Package)* diz respeito à interoperabilidade entre sistemas de informação para os alunos (*LIS – Learner Information Systems*), baseados na Internet, e outros sistemas que suportem um ambiente de aprendizagem, também na Internet.

O objectivo desta especificação é a definição de um conjunto de “pacotes” que possam ser usados para exportar dados para um servidor de informação sobre alunos, e para extrair dados de um tal servidor, pressupondo que este seja IMS compatível.

As estruturas nucleares do IMS LIP são baseadas em: acessibilidades, actividades, dependências, competências, objectivos, identificações, interesses, qualificações, certificações e licenças, relacionamentos, chaves de segurança e transcrições.

### **IMS Question & Test Interoperability Specification**

A especificação IMS QTI (*Question & Test Interoperability*) descreve um modelo de dados destinado à representação de perguntas e testes de avaliação de conhecimentos. Desta forma, a especificação possibilita a troca destes itens entre ferramentas de autoria, bases de dados de questões, ferramentas de geração de testes, sistemas de aprendizagem e sistemas de avaliação automática.

O modelo de dados associado é descrito de forma abstracta, utilizando UML, como forma de disponibilizar um conjunto vasto de representações que podem ser aproveitadas para desenvolvimento, de acordo com as especificidades de produção de cada produtor de software. Contudo, com o propósito de promover a troca entre diferentes sistemas, é fornecido e fortemente recomendado que seja utilizado o formato normalizado XML.

A especificação foi arquitectada de forma a suportar tanto a interoperabilidade como a evolução, através do estabelecimento de pontos de extensão bem definidos.

A documentação sobre esta especificação, na sua versão 2.1, encontra-se distribuída pelos seguintes documentos:

- *IMS Question & Test Interoperability Overview;*
- *IMS Question & Test Interoperability Assessment Test, Section, and Item Information Model;*
- *IMS Question & Test Interoperability XML Binding;*
- *IMS Question & Test Interoperability Results Reporting;*
- *IMS Question & Test Interoperability Implementation Guide;*
- *IMS Question & Test Interoperability Integration Guide;*
- *IMS Question & Test Interoperability Conformance Guide;*

- *IMS Question & Test Interoperability Meta-data and Usage Data;*
- *IMS Question & Test Interoperability Migration Guide.*

### **IMS Resource List Interoperability**

A especificação IMS RLI (*Resource List Interoperability*) detalha a forma como metadados estruturados podem ser trocados entre sistemas que armazenam e disponibilizam recursos, com o propósito de criar listas e de as organizar com objectivos educacionais. Um exemplo típico deste tipo utilização são as listas de leituras aconselhadas em determinados contextos de aprendizagem.

A especificação está baseada no comportamento de um serviço abstracto e num modelo de dados que descreve em termos genéricos um recurso, ao nível do item e ao nível de uma lista de itens, bem como os comportamentos associados aos serviços de gestão dessas listas. O modelo de dados é descrito em XML, combinando elementos que primariamente estabelecem uma relação com subconjuntos de normas IEEE-LOM<sup>36</sup> e ISO 690-2<sup>37</sup> sobre citações bibliográficas destinadas a descrever itens (recursos) e listas de agregação desses itens.

A documentação sobre esta especificação encontra-se distribuída pelos seguintes documentos:

- *IMS Resource List Interoperability Briefing;*
- *IMS Resource List Interoperability Best Practice and Implementation Guide;*
- *IMS Resource List Interoperability XML/WSDL Binding;*
- *IMS Resource List Interoperability Information Model;*
- *IMS Resource List Interoperability Conformance Requirements;*

### **Sharable State Persistence**

A especificação *Sharable State Persistence* descreve uma extensão aos sistemas *eLearning*, que permite o armazenamento e o acesso partilhado a informação de estado

---

<sup>36</sup> Mais informação sobre esta norma em: <http://ltsc.ieee.org/wg12/>

<sup>37</sup> Mais informação sobre a norma em: <http://www.collectionscanada.ca/iso/tc46sc9/standard/690-2e.htm>

entre objectos de conteúdo e, na sua versão 1, foi aprovada pelo *IMS Technical Board* em Julho de 2004.

Presentemente não existe nenhum método recomendado que estabeleça o modo como um objecto de conteúdo deva armazenar esse tipo de informação durante o funcionamento de um sistema de eLearning, de forma a permitir que esse ou outro objecto de conteúdo possa posteriormente aceder a tal informação. Contudo, essa funcionalidade é importante para garantir a persistência da informação de estado que vai sendo gerada por um conjunto variado de conteúdos interactivos.

### **IMS Simple Sequencing Specification**

A especificação *IMS Simple Sequencing* foi tornada pública em Março de 2003 e define um método para representação do que deve ser o comportamento de uma experiência de aprendizagem, desenhada por forma a que qualquer LTS (*Learning Technology System*) possa estabelecer uma sequência de actividades autónomas de aprendizagem, de uma forma coerente. A especificação define as funcionalidades e os comportamentos requeridos de forma a que possam ser implementados sistemas em conformidade com a especificação. Esta inclui regras que descrevem a sequência de acções associadas aos conteúdos, de acordo com os resultados das interacções dos alunos com os mesmos.

### **Vocabulary Definition Exchange**

A especificação *IMS VDEX (Vocabulary Definition Exchange)* define uma gramática para regular a troca de listas de termos, isto é, de vocabulários. A especificação pode ser utilizada, por exemplo, para exprimir informação válida a utilizar em instâncias de IEEE LOM, *IMS Metadata*, *IMS Learner Information Package*, ADL SCORM, etc.

Esta especificação foi aprovada pelo *IMS Technical Board* em Fevereiro de 2004.

### **Abstract Framework**

A *IMS IAF (IMS Abstract Framework)* é um conjunto de regras destinado a permitir descrever o contexto da evolução dos trabalhos do consórcio IMS no desenvolvimento de

especificações relacionadas com as tecnologias de eLearning. Não se trata de uma definição da arquitectura IMS. É antes, um agrupamento de linhas orientadoras destinadas a definir o conjunto de serviços para os quais o consórcio IMS pode ou não vir a desenvolver especificações de interoperabilidade.

Nos casos em que o IMS não produz especificações próprias, todo o esforço é encaminhado para a adopção ou recomendação de especificações de outros organismos de reconhecida relevância internacional.

### **Tools Interoperability Guidelines**

A abordagem IMS TI (*Tools Interoperability*) está relacionada com a crescente procura de um mecanismo reutilizável que permita a integração em plataformas LMS de componentes desenvolvidos por terceiros.

Tais componentes podem adicionar “funcionalidades especialistas” aos LMS, tais como mecanismos de avaliação, ou sistemas de ajuda ao ensino de matérias específicas.

### **Learning Design Specification**

A especificação *IMS Learning Design* suporta um vasto leque de abordagens pedagógicas nos processos de aprendizagem a distância, uma vez que disponibiliza uma linguagem flexível e genérica, em vez de implementar de uma forma rígida, aspectos de um ou mais métodos pedagógicos.

A versão 1 desta especificação foi aprovada pelo *IMS Technical Board* em Fevereiro de 2003.

Dado tratar-se de um projecto relativamente recente, iniciado em 2002, mas em franca expansão, e uma vez que se trata de uma norma que poderia ter sido escolhida, alternativamente, como referência para o trabalho realizado no âmbito desta tese, e ainda porque nesse contexto foi objecto de um estudo mais aprofundado, serão apresentados de uma forma mais detalhada os aspectos achados mais relevantes, relativos ao projecto em causa.

No âmbito do consórcio IMS o projecto *Learning Design* é um projecto transversal a todas as camadas de um LMS e, embora tendo sido iniciado o seu desenvolvimento sobre a especificação *IMS Content Package*, teve a sua origem fora do referido consórcio.

Apesar de se tratar de um projecto recente, tem vindo a verificar uma evolução grande e uma cada vez maior adopção como referencial de desenvolvimento de plataformas de eLearning e de editores de pacotes de conteúdos de aprendizagem.

As plataformas RELOAD - *Reusable eLearning Object Authoring and Delivery* (Reload, 2004) e CopperCore (Kraan, 2004), bem como o editor de pacotes de conteúdos de aprendizagem eXe – eLearning XHTML editor (AIT, 2004), são alguns exemplos de adopção desta especificação.

Ao mesmo tempo, outras plataformas não desenvolvidas de raiz de acordo com a especificação, como é o caso do LAMS (*Learning Actividade Management System*) ou do MOODLE (*Modular Object-Oriented Dynamic Learning Environment*), encontram-se em evolução no sentido da adopção desse referencial ou da compatibilização com o mesmo, nomeadamente ao nível da reutilização de pacotes de conteúdos de aprendizagem.

O *Learning Design Information Model* é uma extensão da especificação *IMS Content Packaging* e integra o trabalho realizado sobre a EML (*Educational Modeling Language*), submetido ao *Learning Design Working Group* pela Universidade Aberta da Holanda, integrando ainda as especificações *IMS MetaData* e *IMS Simple Sequencing*.

As plataformas desenvolvidas de acordo com as especificações IMS LD têm vindo a ser disponibilizadas em versão *open source*, indicando claramente que o negócio do eLearning deixará de ser a breve trecho focalizado na comercialização das próprias plataformas e sim nos conteúdos. Estes, por sua vez, serão também cada vez mais, criados de acordo com as especificações de normalização preconizadas pelos principais projectos em curso. De entre estes, podem salientar-se o SCORM, o IMS, o AICC, o ARIADNE (*Alliance of Remote Instructional Authoring and Distribution Networks for Europe*), o IEEE LTSC, etc., devendo ser evidenciadas as características de reutilização e interoperabilidade que todos preconizam.

Quer a plataforma LAMS, quer a plataforma MOODLE estão disponíveis em versão *open source*, e a segunda está a afirmar-se fortemente no mercado do eLearning, sendo previsível que venha a assumir um papel de liderança dadas as suas características funcionais, a facilidade de utilização, a possibilidade de incorporação de novas



funcionalidades, a ausência de custos de aquisição e o facto de estar desenvolvida por forma a integrar conteúdos compatíveis com a norma SCORM 1.3 e a norma AICC, ao mesmo tempo que procura também uma aproximação à especificação IMS LD.

Simultaneamente, estão em curso trabalhos exploratórios com vista a uma possível implementação da filosofia IMSD LD por parte das diversas versões dos LMS baseados no MOODLE. Não deve ser esquecido o facto de em torno de um núcleo base MOODLE, ser possível desenvolver módulos/funcionalidades satélite, integrados na plataforma original, constituindo versões ou evoluções desta, de forma a obter configurações adequadas às necessidades específicas de cada contexto.

### **A estrutura do IMS LD**

O conceito nuclear da especificação *Learning Design* é o de que, independentemente do modelo pedagógico, cada pessoa assume uma função no processo de ensino/aprendizagem, tornando-se tipicamente num “Aluno” ou num “Elemento de Suporte”. Nessa função, essa pessoa trabalha no sentido da obtenção de determinados objectivos ou resultados, a partir da realização de actividades de aprendizagem ou de suporte mais ou menos estruturadas, no seio de um determinado ambiente. Um ambiente, por sua vez, consiste no conjunto de objectos de aprendizagem e serviços necessários à execução das diversas actividades (IMS, 2003a). O nível A da especificação IMS LD contém todo o vocabulário nuclear de suporte à vasta diversidade pedagógica existente.

O nível B adiciona “Propriedades” e “Condições” ao nível A, de modo a permitir personalizar e obter sequências de actividades e interacções mais elaboradas, baseadas em portfolios. A separação das “Propriedades” e das “Condições” num “esquema” à parte, constitui um melhoramento ao IMS Simple Sequencing.

O nível C acrescenta ao nível B notificações, não se conhecendo ainda neste momento qualquer implementação considerando este nível.

### **Normas ou especificações utilizadas**

O IMS LD integra um conjunto de outras especificações já existentes, que se apresentam de seguida:

- *IMS Content Packaging*. Como forma de melhor obter uma UoL (*Unit of Learning*), é desejável e recomendável a integração do IMS LD num *Content Package*;
- *IMS Simple Sequencing Specification*. O IMS-SSS pode ser utilizado para determinar a sequência de recursos dentro de um LO (*Learning-Object*) e para determinar a sequência de diferentes LO e serviços dentro de um ambiente;
- *IMS/LOM Metadata*;
- *IMS-QTI (Question and Test Interoperability)*;
- *IMS-RDCEO (Reusable Definition of Competency or Educational Objective)*;
- *IMS-LIP (Learner Information Package)*;
- *IMS Enterprise*. Pode ser utilizado para associar “Alunos” e “Elementos de Suporte” a *roles*, aquando da instanciação de um *learning design*.

Como se verá com maior detalhe, mais adiante nesta secção, o IMS Learning Design assenta o seu funcionamento na especificação de *learning designs*, de *plays* e de *conditions*, sendo que um *learning design* é uma descrição de um método possibilitando que alunos alcancem determinados objectivos ao realizarem determinadas actividades de aprendizagem, segundo uma ordem definida no contexto de um determinado ambiente de aprendizagem.

De acordo com a especificação IMS LD, é possível incluir conteúdos SCORM num *learning design*. Este aspecto introduz uma questão curiosa. É suposto o projecto SCORM integrar participações de diversas proveniências, nomeadamente do projecto IMS. Tal facto deveria fazer da norma um todo maior que qualquer das partes. Contudo, como se disse, o IMS LD, prevê a inclusão de conteúdos SCORM, embora o inverso não se verifique. Este facto parece dar força aos que afirmam que o projecto IMS *Learning Design* está a avançar a um ritmo superior àquele que se regista em relação ao SCORM.

A figura 4 apresenta os níveis de agregação semântica no nível C do *Learning Design*, utilizando uma notação UML 1.4 e representando apenas relações de agregação (incluindo composições) e especializações de classes abstractas.

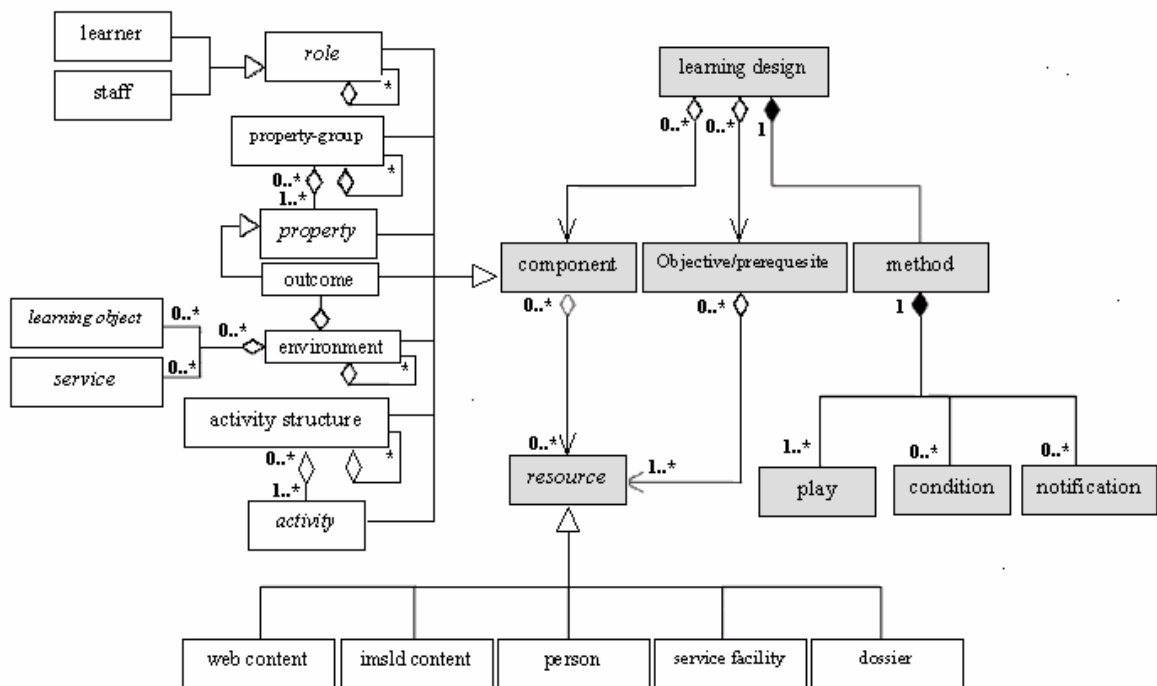


Figura 4 - Agregação semântica do Learning Design no nível C.  
Adaptado de IMS Learning Design Information Model (IMS, 2003a).

De notar que no âmbito do IMS LD, os recursos são considerados como um todo, sejam recursos materiais, tecnológicos ou humanos, numa generalização que, contudo, aparece decomposta nas especializações que o diagrama da figura 4 apresenta.

A figura 5 é uma outra representação conceptual ou modelo do *IMS Learning Design*.

Importa referir que o conceito de notificação utilizado no âmbito do IMS LD se destina a inicializar o processo segundo o qual determinado actor (*role*) é solicitado em determinado momento para desempenhar certas actividades. Esta função pode também ser desempenhada pelo método (*method*). Por exemplo, quando um aluno completa uma actividade (*outcome*), então outro aluno ou o professor pode ser notificado e ver-lhe ser atribuída outra actividade.

Uma notificação é originada por um determinado resultado (*outcome*) e pode provocar a activação de uma nova actividade a ser desempenhada por uma determinada função (*role*).

Este conceito não é novo, podendo referir-se um mecanismo de *workflow*, proposto em (Pinto, 1998), relativo ao relacionamento entre “Aluno”, “Professor” e “Revisor”, num

processo de emissão de novos pedidos de trabalho ao aluno, apenas após este conhecer o resultado do seu anterior desempenho.

Nesse contexto, o “Professor” solicita um trabalho ao “Aluno”. Após a sua realização, este submete-o ao “Revisor”, eventualmente coincidindo fisicamente com o “Professor”. O “Revisor” ao devolver ao “Aluno” o trabalho anotado/classificado, provoca a notificação automática do “Professor”, criando as condições para que este possa gerar uma nova solicitação ao “Aluno”.

Existe uma diferença significativa entre os objectivos das notificações previstas no *Learning Design* e aqueles que se lhes atribuiu no âmbito desta tese. Com efeito, as notificações propostas neste trabalho foram identificadas como um processo necessário para permitir corrigir percursos ou situações anormais (desvios face ao previsto no plano do curso), durante a execução das actividades de um curso.

Pode dizer-se que o modelo proposto neste trabalho apresenta uma aparente maior rigidez, quando comparado com o que é proposto no projecto IMS LD, já que o mesmo assenta num processo de execução de actividades com inícios e fins previamente definidos.

Em todo o caso, se no modelo proposto fossem introduzidas as características disponibilizadas pelo IMS LD, que permitem alterar dinamicamente a configuração de um curso, obter-se-ia ainda assim, uma estrutura composta de um conjunto de actividades que pode perfeitamente ser enquadrada pelo modelo que se propõe no presente trabalho. Isto é, seja qual for o encaixe temporal de uma actividade, esta pode ser sempre precedida de um aviso e, caso não seja executada dentro da janela temporal predefinida, provocará o envio de um alarme. Apenas um formato de curso segundo o qual a execução das diversas actividades fosse puramente aleatória criaria algumas dificuldades no funcionamento do modelo que aqui se propõe. Mas mesmo nesse caso, haveria uma solução. O conjunto de todas essas actividades seria uma “macro actividade”, esta sim, com possibilidade de aviso prévio e de alarmes, quer antes de atingido o fim do curso (primeiro alarme), quer na situação em que todo o conjunto das actividades não fosse executado dentro do tempo previsto (segundo alarme).

No *Learning Design* um método (*method*) consiste numa ou mais sessões (*plays*) concorrentes. Por sua vez, uma sessão é formada por um ou mais actos (*acts*) sequenciais e um acto está associado com uma ou mais componentes de função (*role-parts*). Cada

componente de função associa exactamente uma função a uma actividade (*activity*) ou estrutura de actividades (*activity-structure*).

A leitura simplificada que se pode efectuar deste modelo permite por outro lado identificar que uma pessoa desempenha determinada função e que no âmbito dessa função (*role*) são realizadas determinadas actividades no seio de uma plataforma ou ambiente (*environment*) específico. Este ambiente, por sua vez, é constituído por um conjunto de objectos de aprendizagem (*learning objects*) e/ou serviços.

learning design

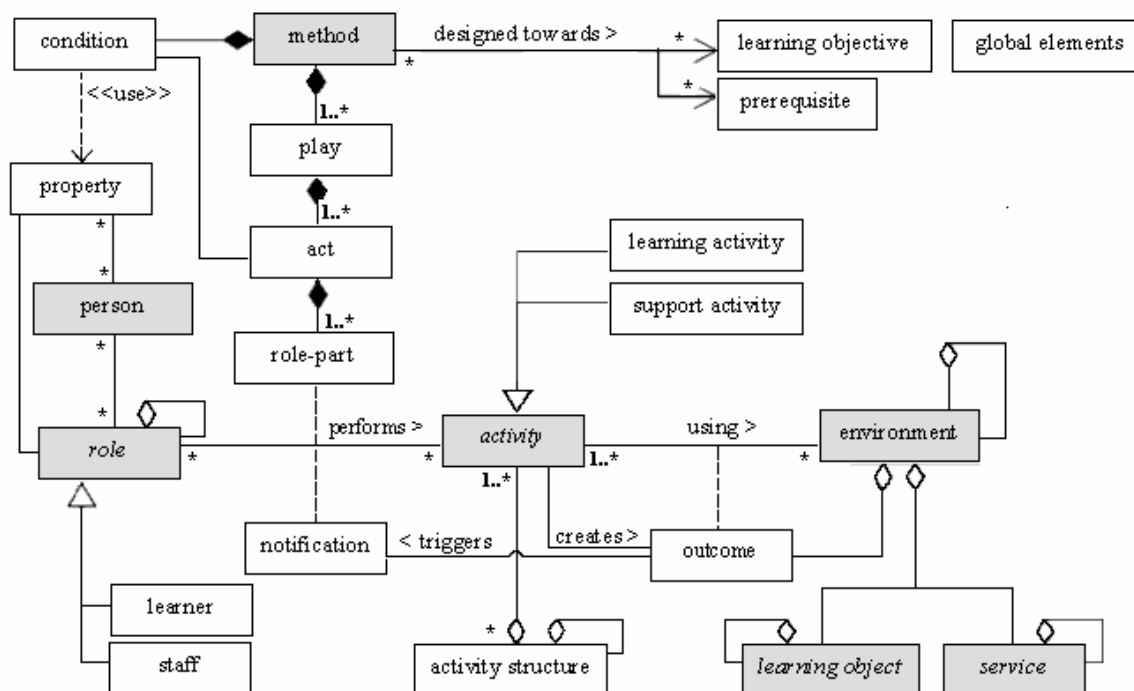


Figura 5 – Modelo conceptual do Learning Design.  
Adaptado de IMS Learning Design Information Model (IMS, 2003a).

No âmbito do IMS LD, um ambiente pode conter dois tipos básicos de objectos:

- a) **Objectos de Aprendizagem Localizados (Recursos)**. Em regra estes LO são especificados por um URL que indica a sua localização, com metadados opcionais associados. Em EML os objectos de aprendizagem são classificados nos seguintes três tipos: Objectos de Conhecimento (*Knowledge-Objects*), Objectos de Ferramentas (*Tool-Objects*) e Objectos de Teste (*Test-Objects*).

b) **Serviços Genéricos.** Um serviço está relacionado com uma qualquer facilidade disponível durante o tempo de execução. Como exemplos de serviços contemplados pelo IMS LD podem identificar-se os seguintes: fóruns de discussão, salas de *chat*, ferramentas de monitorização, funcionalidades de pesquisa, etc.

Dependendo do que seja considerado pelo IMS LD, e isso não está claro na documentação existente, as ferramentas de monitorização podem coincidir total ou parcialmente com a gestão de processos de *eLearning* em tempo real proposta neste trabalho.

Tal como já referido, um *learning design* é uma descrição de um método possibilitando que alunos alcancem determinados objectivos ao realizarem determinadas actividades de aprendizagem, segundo uma ordem definida no contexto de um determinado ambiente de aprendizagem.

No nível A do IMS LD, está previsto um elemento relacionado com a execução de uma actividade (*complete-activity*) que pode ser actualizado quando é atingido o limite temporal predefinido para a actividade. Este elemento coincide claramente com o previsto no modelo que se propõe no âmbito do presente trabalho.

No contexto do IMS LD, um Objecto de Aprendizagem é qualquer recurso, digital ou não digital, endereçável e reproduzível, utilizado na execução de uma qualquer actividade de aprendizagem ou de suporte.

Ao contrário do que acontece com os Recursos, não é possível associar URL aos Serviços durante a fase de Design. Têm que ser instanciados em cada utilização específica, caso contrário, todas as utilizações ou instâncias teriam que fazer uso exactamente dos mesmos serviços predefinidos (*chats*, fóruns, etc).

Dois dos elementos mais importantes da especificação *Learning Design* são o *play* e o *condition*.

A parte nuclear do *Learning Design* é representada no *play*. É aqui que se especifica o *learning design* actual, o processo de ensino/aprendizagem e que funções (*roles*) devem desempenhar que actividades, e por que ordem.

Um *play* consiste numa série de acções, cada uma delas composta por um conjunto de *role-parts*.

Os elementos *time-limit* e *on-completion* definem o que fazer quando é atingido o limite temporal para uma unidade de aprendizagem (UoL).

Deve ter-se presente que segundo o *IMS Learning Design* uma “Unidade de Aprendizagem” é definida como sendo:

$$UoL (Unit of Learning) = IMS Content Package + IMS Learning Design$$

Para criar uma UoL (*Unit of Learning*), é necessário incluir num *IMS Content Package*, um *learning design* dentro do elemento *organizations*.

O excerto de código XML a seguir, mostra a forma como estas diferentes entidades estão relacionadas numa estrutura perfeitamente coerente, de acordo com o que se acabou de referir.

```
<manifest>
  <metadata/>
  <organizations>
    <learning-design xmlns="[standard-namespace-for-learning-design]">
      [add learning design elements here]
    </learning-design>
  </organizations>
  <recursos/>
</manifest>
```

O IMS LD apresenta uma estrutura hierárquica que se inicia com o elemento *method*, se prolonga no elemento *play* e este, por sua vez, no elemento *act*, podendo encontrar-se os elementos *time-limit* e *on-completion* em cada um destes níveis.

O modelo de informação do IMS LD, no nível C, contempla um serviço de envio de mensagens de correio electrónico aos diferentes tipos de utilizadores e, assim, esta funcionalidade, juntamente com os elementos *time-limit* e *on-completion*, podiam ser utilizados para tratar o problema que esteve na base da realização deste doutoramento.

No nível B da especificação, o elemento *conditions* introduz a possibilidade de ser executada uma acção, dependente do teste de uma condição. Tal acção pode ser executada no nível B do *Learning Design*, podendo ser do tipo “mostrar”, “esconder”, “alterar\_estado\_propriedade” ou no nível C, através de uma operação de notificação *notify* dirigida a uma determinada “função” (*role*).

No IMS LD uma notificação pode ter como destinatário, um actor ou um componente do sistema.

No âmbito daquilo que se propõe no presente trabalho o destinatário de uma “notificação” deverá ser sempre a camada de *messaging* do sistema (cf. figura 33, página 129), a quem deve ser passado o conteúdo da notificação, bem como a identificação do destinatário ou destinatários da mesma, e eventualmente, o meio preferencial de comunicação com cada um dos destinatários (endereço de correio electrónico, nº de telefone, endereço postal, etc). O meio preferencial de comunicação pode ser passado à camada de *messaging* ou não, uma vez que se trata de um atributo (“Actor\_notificação\_tipo”) da entidade “actor”, cujo preenchimento é obrigatório (cf. tabela 4, página 145).

Assim sendo, a camada de *messaging* pode, com base no identificador do destinatário da notificação, obter o meio preferencial de comunicação definido para este.

Esta camada pode ser considerada como fazendo parte do sistema global (LMS), ou seja, todo o utilizador, independentemente do seu papel no processo de ensino/aprendizagem, deverá interagir com um único artefacto tecnológico.

Mesmo nos casos em que para comunicar assincronamente fosse necessário utilizar um sistema de correio electrónico não acoplado ao LMS, poder-se-ia pensar num mecanismo da plataforma que guardasse a informação relevante, antes de a passar ao cliente de correio electrónico utilizado especificamente por um dado utilizador.

O nível C do IMS LD adiciona a possibilidade de o *learning designer* especificar o envio de mensagens e de atribuir novas actividades, em função da ocorrência de certos eventos, sendo que o único tipo de notificação previsto é a mensagem de correio electrónico.

## **Modelo Comportamental**

O Modelo Comportamental do *IMS Learning Design* é constituído de duas partes principais: *Instantiation* e *Runtime*.

A componente *Instantiation* é subdividida em:

- *Instantiation Roles* – Com a função de especificar quem está associado às *roles*, já que um *learning design* apenas permite especificá-las. Esta associação é a parte principal da instanciação de um *learning design*;



- *Instantiation Services* – Uma vez conhecidas as funções (*roles*), podem ser definidos serviços genéricos para os actuais membros dessas *roles*, de acordo com as regras associadas à utilização do serviço e levando em linha de conta as permissões definidas para cada *role*.

A componente *Runtime* cobre os aspectos comportamentais de uma UoL nos três níveis de um Learning Design, sendo que no nível A são tratados aspectos relativos ao sequenciamento e à selecção das estruturas de actividades (*Activity-structure*), bem como à gestão da execução de uma UoL, a cargo do elemento *method*.

No nível B são abordados aspectos relacionados com as condições. Todas as condições são precondições definidas aquando do *design time* e devem ser avaliadas durante a fase de execução.

Relativamente ao nível C, é definida a gestão das notificações eventualmente despoletadas por eventos que são do conhecimento do ambiente de execução (*runtime environment*) e que podem materializar-se em actividades concluídas, avaliação verdadeira de expressões ou actualização do valor de certos atributos.

No âmbito do presente trabalho, não houve qualquer preocupação com os mecanismos de acompanhamento da execução das actividades e da respectiva actualização do “estado” dessas mesmas actividades. Isto é, se foram terminadas ou não. De igual modo, não houve nenhuma preocupação com o resultado da execução dessas actividades, isto é, se o resultado de uma determinada actividade correspondia ao que era esperado pelo autor do curso. Apenas se fez incidir a atenção no processo de identificação da execução das actividades dentro das janelas temporais previstas no plano do curso. Foi assumido que qualquer outro tipo de “cuidados” seria da responsabilidade das outras camadas dos LMS.

### 3.3 O projecto AICC

#### 3.3.1 Introdução

A AICC é uma associação internacional de profissionais vocacionados para o ensino/treino baseado em tecnologias. Produz recomendações e normas para a indústria

aeronáutica, no que toca ao desenvolvimento, distribuição e avaliação de CBT e tecnologias de treino associadas.

Existindo desde 1988, foi originalmente criada para conceber normas de hardware constituindo plataformas de CBT. Posteriormente ramificou-se em diversas outras áreas de actividade.

As recomendações AICC são suficientemente genéricas para serem adoptadas num significativo número de tipos de sistemas de treino baseados em computador (CBT) e, por esse motivo, são largamente utilizadas fora do âmbito das acções de treino da indústria aeronáutica (AICC, 2006).

O consórcio AICC também coordena activamente os seus esforços com outros organismos empenhados no desenvolvimento de normas, tais como o IMS, a ADL, a ISO SC/36 e o IEEE/LTSC.

Curiosamente, o consórcio AICC não desenvolve nem comercializa *courseware* orientado para o treino na área em que é especialista, embora alguns dos seus membros o façam.

Desse grupo fazem parte construtores de aviões, organizações vocacionadas para o treino em aviação (militar, civil e comercial), agências governamentais reguladoras de política aeronáutica, empresas de desenvolvimento de software e produtores de CBT e *courseware*.

Alguns dos membros da AICC participam nos trabalhos quer do consórcio, quer da ATA (*Airline Transport Association*)<sup>38</sup> para coordenar actividades relacionadas com o treino. Contudo, o comité concentra os seus esforços nos materiais de treino interactivo, enquanto que a ATA está mais vocacionada para a produção de manuais, seja em formato electrónico, seja em papel.

O consórcio desenvolve recomendações técnicas referenciadas como AGR (*AICC Guidelines & Recommendations*) e respectivas especificações.

Um sistema para formação/treino é classificado como estando em conformidade com o AICC se estiver conforme com pelo menos um dos AGR. Dada a existência de um significativo número de AGR, isso significa que um tal sistema pode ser compatível com o AICC a vários níveis. A situação mais comum de conformidade está relacionada com a verificação das recomendações constantes dos documentos AGR-006 (*File-based CMI*

---

<sup>38</sup> <http://www.airlines.org/>

*systems*) ou com o AGR-010 (*Web-based CMI Systems*). Estes AGR definem a comunicação entre sistemas CMI e *courseware* CBT.

### 3.3.2 Objectivos

De acordo com a informação disponível no sítio oficial do consórcio<sup>39</sup>, os objectivos do AICC são os seguintes:

- Apoiar os operadores da indústria aeronáutica no desenvolvimento de linhas orientadoras capazes de possibilitar uma efectiva e económica implementação de conteúdos para treino assistido por computador (CBT);
- Desenvolver linhas orientadoras capazes de permitir interoperabilidade;
- Promover um fórum aberto para a discussão da utilização de tecnologias de treino, seja sobre CBT ou outras.

### 3.3.3 Normas, especificações ou recomendações geradas

Tal como se referiu na secção 3.3.1, o consórcio AICC desenvolve recomendações técnicas identificadas pela sigla AGR<sup>40</sup>.

Cada AGR faz recomendações técnicas numa área específica. Por exemplo, a AGR-002 é a recomendação para o hardware constituindo uma estação de trabalho. Estas recomendações são breves na maioria dos casos e, se há necessidade de um maior detalhe, o AGR faz referência a um ou mais documentos técnicos do consórcio.

O AICC possui 11 AGR (AGR-002 a AGR-012). A recomendação AGR-001 lista todas as outras AGR e não é considerada uma recomendação ela própria.

Apresentam-se seguidamente algumas das principais recomendações (AGR) efectuadas pelo consórcio:

- AGR-002 - 1989 – Plataforma comum de recomendações para o desenvolvimento de CBT

---

<sup>39</sup> <http://www.aicc.org/>

<sup>40</sup> Informação sobre os AGR disponível em <http://aicc.org/pages/down-docs-index.htm#AGR>.

- AGR-003 - 1992 – Uma especificação em DOS para áudio digital, antes do aparecimento das normas multimédia para Windows. Esta especificação permitia que os utilizadores finais pudessem utilizar um único circuito áudio, compatível com *courseware/CBT* proveniente de diferentes fornecedores. Devido ao elevado número deste tipo de software, que foi sendo desenvolvido ao longo dos tempos, ainda hoje esta especificação se encontra activa.
- AGR-004 – 1999 - O objectivo desta recomendação consiste em possibilitar que uma empresa de aviação possa integrar num sistema de treino, *courseware CBT* de larga utilização, ao mesmo tempo que considera outros três factores:

1. **Flexibilidade.** São feitas recomendações no sentido de os sistemas permitirem a utilização de *courseware* alternativo, ao mesmo tempo que o próprio sistema de suporte deve poder ser configurado de acordo com as necessidades específicas de determinado cliente.

2. **Expansibilidade.** Devem ser salvaguardadas evoluções do *courseware* que impliquem o *upgrade* das componentes *hardware* e *software*. Esses *upgrades* devem poder ser realizados ao menor custo possível.

3. **Mudança de tecnologias.** O consórcio deverá preocupar-se de uma forma continuada com o surgimento de novas tecnologias que possam ser adequadas ao melhoramento do CBT. Desta forma, estas recomendações terão tendência para uma actualização periódica.

- AGR-005 – 1992 – Esta recomendação propõe linhas orientadoras no sentido da promoção de interoperabilidade de dispositivos periféricos tais como:
  - Placas vídeo multi-função;
  - Reprodutores de vídeo;
  - Écrans tácteis, ratos ou *trackballs*.

Neste contexto a interoperabilidade é vista como a capacidade de executar *courseware* em diferentes sistemas, dotados de diferentes dispositivos periféricos, bem como a possibilidade de utilizar cursos interactivos suportados por vídeo, provenientes de fornecedores diversos.

- AGR-006 - 1993 – Recomendação para interoperabilidade em CMI, envolvendo LMS. Esta recomendação/especificação resultou no desenvolvimento de sistemas CMI capazes de partilhar informação com *courseware* CBT funcionando em cima de redes locais de computadores (*LAN – Local Area Network*), com origem em diversos fabricantes.
- AGR-007 – 1995 – Esta recomendação fornece orientações para o intercâmbio de elementos que podem ser encontrados no âmbito do *courseware*, tais como:
  - Texto
  - Gráficos
  - Vídeo
  - Áudio
  - Componentes software

O propósito desta recomendação é facilitar a migração de *courseware*, de um sistema para outro.

- AGR-008 – 1995 – Esta recomendação fornece linhas orientadoras para a criação, distribuição e utilização de vídeo digital em *courseware* CBT.
- AGR-009 - 1996 – Especificação de linhas orientadoras para a criação de ícones normalizados, a utilizar na navegação feita nos CBT. O objectivo é simplificar e uniformizar os interfaces, por forma a que a atenção dos alunos/utilizadores não saia prejudicada ao terem que interagir com diferentes sistemas.
- AGR-010 - 1998 – As especificações dos sistemas CMI foram actualizadas de forma a incluir CBT ou WBT (*Web Based Training*), dando origem a esta recomendação.
- AGR-011 - 2005 – É lançada a especificação do PENS (*Package Exchange Notification Services*) que possibilita que sistemas de

autoria/gestão de conteúdos integrem o processo de publicação no seio de LMS.

- AGR-012 - 2005 – É apresentada uma *checklist* de linhas orientadoras AICC que devem ser consideradas antes de desenvolver ou comercializar conteúdos ou sistemas CBT/WBT. A descrição dos elementos dessa *checklist* é feita no documento *Training Development Checklist*<sup>41</sup>.

Para além das recomendações que se acabou de apresentar, o consórcio desenvolveu procedimentos de teste para certificação formal de conformidade com as recomendações AGR-006 e AGR-010 e, presentemente, oferece testes de certificação para sistemas CMI e *courseware* CBT<sup>42</sup>.

## 3.4 O projecto ARIADNE

### 3.4.1 Introdução

A Fundação ARIADNE<sup>43</sup> foi criada para explorar e desenvolver os resultados dos projectos Europeus ARIADNE e ARIADNE II, terminados em 2002, que haviam criado ferramentas e metodologias para produção, gestão e reutilização de componentes pedagógicas passíveis de ser incluídas em sistemas baseados em computador e de *curricula* de treino suportados telematicamente.

A validação das ferramentas e dos conceitos teve lugar em diversas instituições académicas e empresariais por toda a Europa e a fundação foi encorajada a prosseguir com a ideia da exploração não comercial.

Os resultados obtidos são consequência de um grande esforço em R&D (*Research and Development*), aproximadamente 100 homens ano investidos até Junho de 2000, uma grande parte suportada financeiramente pela União Europeia e pelo governo suíço.

---

<sup>41</sup> O documento pode ser consultado em <http://aicc.org/docs/AGRs/agr012v1.pdf>

<sup>42</sup> Informação acerca da conformidade com o AICC disponível em <http://www.aicc.org/pages/primer.html>.

<sup>43</sup> <http://www.ariadne-eu.org/>

Segundo a fundação, a manutenção e o treino continuados no uso das melhores ferramentas do projecto ARIADNE, assim como seu desenvolvimento complementar, consolidarão e aumentarão a comunidade de utilizadores que emergiu em torno do projecto.

### 3.4.2 Objectivos

A Fundação ARIADNE é uma associação sem fins lucrativos, que procura atingir um pequeno número de objectivos sociais básicos:

- Encorajar a cooperação entre instituições educacionais através do lançamento e exploração de um verdadeiro repositório de conhecimentos Europeu;
- Garantir que a educação continuará a ser dominada por aspectos sociais e de cidadania, combatendo a tendência para fazer dela um mero objecto de mercado;
- Suportar e proteger a diversidade linguística e o uso das línguas maternas, nacionais ou regionais, no processo educativo;
- Definir através de um consenso internacional quais os aspectos da formação baseada em TIC (Tecnologias da Informação e da Comunicação) que devem ser normalizados e quais aqueles que devem ser deixados com as características locais ou nativas;

De acordo com a informação disponível no sítio oficial<sup>44</sup>, a fundação procura incrementar a tomada de consciência por parte dos cidadãos aprendentes europeus, da existência de canais de treino baseados em TIC, convencer e aconselhar novos potenciais utilizadores da comunidade académica (principalmente instituições do sector público), e assistir de uma forma crescente novos utilizadores do mundo empresarial, onde o treino e a reciclagem desse treino se revelem necessários.

Esta associação implementa e promove abordagens colaborativas, no âmbito das quais as instituições de ensino e as entidades empresariais são incentivadas a cooperar no sentido

---

<sup>44</sup> <http://www.ariadne-eu.org/>

da optimização das suas tarefas de treino, em vez de entrarem numa competição cega entre si.

O princípio da partilha e da reutilização é reflectido na tecnologia desenvolvida no âmbito do projecto ARIADNE, disponibilizada para todos os membros da fundação.

A associação trabalha para uma alargada adopção de tecnologias educacionais por parte da sociedade europeia, tentando a adopção de práticas independentes da plataforma tecnológica utilizada. Isto envolve uma abordagem que tem em conta as sensibilidades nacionais e regionais e as preferências em tudo o que diz respeito à educação. A abordagem e as ferramentas ARIADNE, dado o seu carácter pedagógico genérico e neutral, são um bom ponto de partida para alcançar esses propósitos.

As seguintes tecnologias e metodologias educacionais estão à disposição dos membros da Associação ARIADNE:

- Ferramentas para indexação multilíngua de objectos de aprendizagem;
- Capacidade de partilha e reutilização de objectos de aprendizagem;
- Ferramentas de autor para a criação de objectos de aprendizagem;
- Ferramentas de captura de informação sócio-geográfica dos alunos;
- Desenho de *curricula* sócio-geograficamente orientados;
- Selecção e agregação de objectos de aprendizagem, em *curricula* específicos;
- Desenho de cursos a distância distribuídos na Web;
- Boas práticas na utilização de tecnologias de comunicação interactivas;
- Boas práticas na gestão de pequenos, médios e grandes cursos em modelo ODL (*Open and Distance Learning*).

## 3.5 O projecto ADL SCORM

### 3.5.1 Introdução

O projecto SCORM iniciou-se em 1997 no seio da iniciativa ADL, lançada pelo Departamento de Defesa Norte-Americano, tendo a versão 1.0 ficado disponível em Janeiro de 2000. Desde então, a norma tem sofrido a evolução que a figura 6 representa.



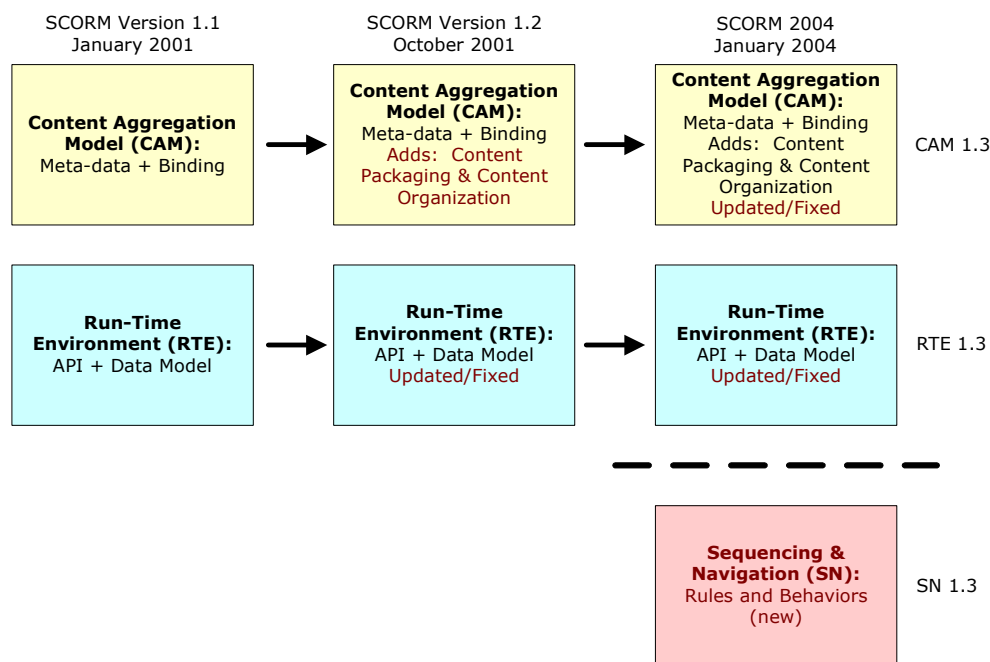


Figura 6 – Evolução do projecto SCORM.  
Adaptado de (ADL, 2004a).

A norma é o resultado da adopção/inclusão dos contributos provenientes de diversas organizações e consórcios com trabalho na área, por forma a criar uma norma mais geral para o desenvolvimento de plataformas de ensino/aprendizagem a distância sobre a Web, permitindo interoperabilidade e reutilização de conteúdos de aprendizagem.

A ADL tem feito progredir o seu trabalho com base numa colaboração estreita com a indústria e com organismos académicos e governamentais, de forma a disponibilizar especificações e normas abertas, aceites internacionalmente.

Embora tenham sido muitas as organizações a contribuírem ao longo do tempo para o que a norma SCORM é hoje, quatro dessas organizações devem ser destacadas pela qualidade e quantidade das suas participações. Trata-se da AICC, da ARIADNE, do IEEE LTSC e do IMS.

Presentemente na versão SCORM 1.3 ou SCORM 2004, a informação descrevendo a norma está organizada num conjunto de documentos (cf. Figura 1, página 4), cada um dos quais abordando aspectos particulares da arquitectura e do funcionamento globais de um LMS, dos conteúdos e sua construção, da sua organização e sequenciamento e do acesso e navegação que neles pode fazer-se (ADL, 2004a).

Ainda de acordo com a mesma fonte (ADL, 2004a), a norma, cuja descrição é feita nos diversos documentos (livros), fornece uma API para a comunicação de informação relativa às interações dos alunos com objectos de conteúdo, um modelo de dados para representação dessa informação, uma especificação sobre como empacotar conteúdos que possam ser interoperáveis entre diferentes plataformas, um conjunto normalizado de elementos de metadados que podem ser utilizados para descrever conteúdos de aprendizagem, e um conjunto de regras de sequenciamento que podem ser utilizadas para organizar o acesso aos conteúdos de aprendizagem.

### 3.5.2 Objectivos

Na dependência directa de um gabinete associado ao secretário da defesa norte-americano, à iniciativa ADL foram impostos os objectivos de liderar um esforço colaborativo no sentido de aumentar o potencial das tecnologias da informação por forma a modernizar os processos de aprendizagem.

Nesse contexto, a ADL desenvolve esforços colaborativos e adaptativos envolvendo os sectores público e privado, no sentido do desenvolvimento de normas, ferramentas e conteúdos de aprendizagem para os ambientes de ensino/aprendizagem do futuro.

A visão da iniciativa ADL consiste em fornecer os melhores processos de ensino/aprendizagem e a ajuda na obtenção de melhores desempenhos, de acordo com as características e necessidades específicas de cada um, a um custo aceitável, e independentemente da hora e do lugar.

### 3.5.3 Normas, especificações ou recomendações geradas

O resultado do muito trabalho realizado no âmbito do projecto SCORM, que originou a norma com a mesma designação, encontra-se documentado, tal como se referiu na secção 3.5.1, em três livros. O conteúdo de cada um deles será apresentado a seguir.

O primeiro dos documentos é o CAM (*Content Aggregation Model Book*) (ADL, 2004a), (ADL, 2004b). Neste livro é possível encontrar linhas orientadoras para o empacotamento de conteúdos de aprendizagem estruturados. O documento descreve a

nomenclatura para os objectos de aprendizagem e o *SCORM Content Packaging*, referenciando ainda o *IMS Learning Resource Metadata Informação Model*.

De referir que o *IMS Learning Resource Metadata Informação Model* é baseado no LOM do IEEE-LTSC que, por sua vez, foi desenvolvido no âmbito dos projectos IMS e ARIADNE.

O propósito do CAM é proporcionar meios para a produção de conteúdos educativos com base em fontes acessíveis, reutilizáveis, partilháveis e interoperáveis. Adicionalmente, o CAM estabelece a forma como os objectos de conteúdos educativos podem ser identificados e descritos, agregados num curso ou em parte dele, e movimentados entre sistemas, tais como LMS e repositórios. Neste livro é ainda possível encontrar uma completa definição sobre os processos de implementação destas funcionalidades. O modelo inclui especificações para a agregação de conteúdos e definições de metadados (ADL, 2004b).

Um segundo livro descreve o RTE (Run-Time Environment) (ADL, 2004c). Nele é possível encontrar indicações para lançar, comunicar e fazer o seguimento de um objecto de conteúdo (*CO – Content Object*) num ambiente baseado na Web. Este livro é uma evolução do “*Execution Environment*” definido pelas CMI001 (*Guidelines for Interoperability*) do AICC. A ADL colaborou com os membros do AICC para:

- Desenvolver o *Launcher*, um mecanismo para promover o “lançamento” de objectos de conteúdo;
- Especificar uma API;
- Adoptar o “*Data Model for Web-based data elements*” do AICC.

O principal propósito do RTE é implementar um mecanismo de comunicação entre o objecto de conteúdo e o LMS.

Um objecto de conteúdo deverá ser utilizável em múltiplas plataformas, independentemente da ferramenta usada para a sua criação. Para que este requisito se verifique, é necessário que estejam reunidas as seguintes condições:

- Uma forma comum de inicialização do objecto de conteúdo;
- Uma forma comum para a comunicação do objecto de conteúdo com o LMS;
- A identificação dos elementos de dados que devem ser trocados entre o objecto de conteúdo e o LMS, durante o processo de comunicação.

Os três componentes do RTE estão definidos neste livro e consistem no *Launcher*, na API e no “*Data Model*” (SCORM, 2004c), sendo mostrada na figura 7 uma representação conceptual do RTE.

O *Launcher*, como parte da arquitectura de um LMS, tal como é proposto pelo projecto SCORM, é responsável pela inicialização do processo de apresentação de um novo objecto de conteúdo no *browser* do utilizador (normalmente, o aluno).

Estão previstos dois tipos de objectos, SCO (*Shareable Content Object*) e *Asset*, sendo de realçar o facto de apenas existir comunicação entre o LMS e objectos de conteúdo do tipo SCO.

O mecanismo de comunicação é estabelecido com recurso a métodos predefinidos da API.

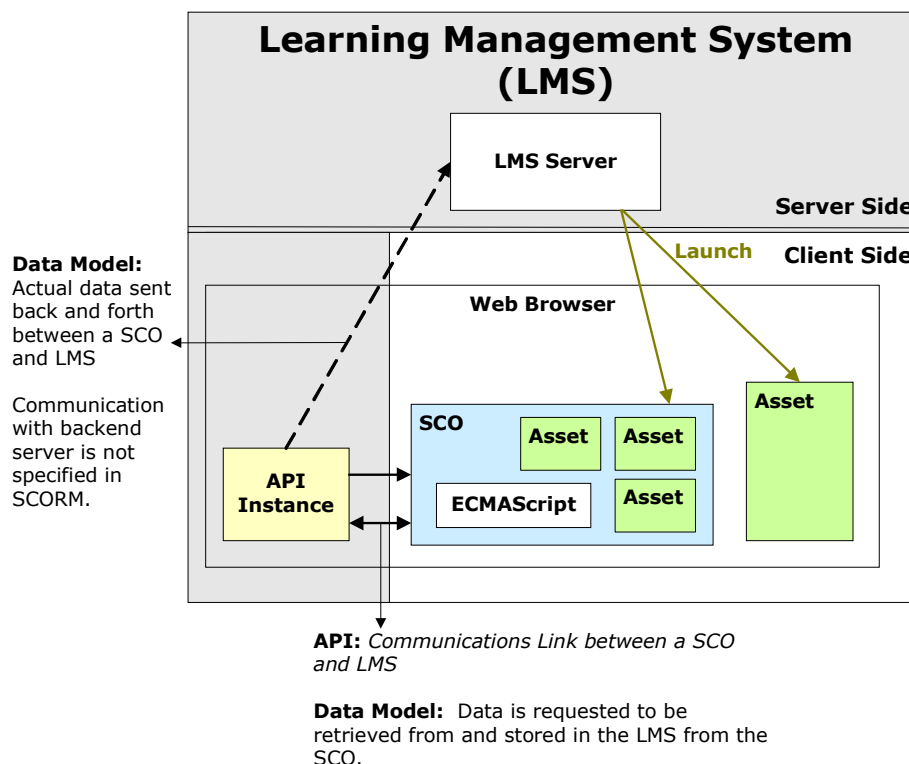


Figura 7 – Representação conceptual do Run-Time Environment SCORM.  
Adaptado de (ADL, 2004c).

A API possui os mecanismos de comunicação (métodos) para informar o LMS do estado da comunicação com um SCO (por exemplo, inicializado, terminado e/ou num estado de erro) e alguns dos seus métodos são usados também para recuperar e armazenar informação durante a interacção do LMS com o SCO.

O *Data Model* é um conjunto normalizado de elementos informacionais, utilizados para caracterizar e registar a informação relativa à evolução da interacção de um SCO com o LMS. Nele são definidos os elementos do modelo a que tanto o LMS como o SCO têm que aceder durante o processo comunicacional entre ambos.

A utilização destes elementos do *Data Model* viabiliza a reutilização e interoperabilidade dos SCO em diferentes sistemas (LMS).

Por outro lado, o aspecto fulcral que justifica todo o trabalho de normalização realizado e a desenvolver, não só no âmbito do projecto ADL SCORM, mas também no dos outros projectos de normalização já aqui referidos, é a interacção dos utilizadores com os LMS e com os cursos neles disponibilizados.

No projecto SCORM apenas os alunos são considerados como actores interagindo com os LMS e os cursos, e um aluno torna-se envolvido com um CO logo que uma actividade tiver sido identificada para ser lançada e o CO correspondente (SCO ou *Asset*) tiver sido apresentado no *browser* do aluno.

O projecto adoptou a norma IEEE P1484.11.1 *Draft Standard for Learning Technology – Data Model for Content Object Communication* (IEEE LTSC, 2004) para definir o processo comunicacional entre os objectos de conteúdo e os LMS durante as interacções dos alunos com os cursos e as plataformas.

Conhecer o significado de alguns termos chave definidos pela norma IEEE P1484.11.1 é fundamental para que se possa perceber esse processo e a necessidade desse conhecimento justifica a descrição que se apresenta de seguida.

- **Learner Attempt** – Registo de uma tentativa por parte de um aluno para satisfazer os requisitos de uma actividade de aprendizagem que usa um CO. Um *Learner Attempt* inicia-se quando o LMS identifica uma actividade a ser apresentada e pode estender-se por várias sessões de utilização.  
Como facilmente se compreenderá, uma actividade pode não ser executada numa só sessão e, portanto, um *Learner Attempt* pode ficar “suspense” sempre que uma sessão for interrompida, antes que a actividade esteja terminada.
- **Learner Session** – É um período de tempo não interrompido, durante o qual um aluno está a aceder a um CO. Inicia-se quando é lançado no *browser* do aluno um CO (SCO ou *Asset*) e pode ser terminado, deixando o CO num estado “suspense” ou “normal”, conforme o aluno tenha atingido ou não os objectivos definidos pelo CO. Se estes tiverem sido atingidos, também o *Learner Attempt* correspondente terminará.
- **Communication Session** – Consiste numa ligação activa entre um SCO e uma instância da API, tendo início quando o CO inicia a comunicação com o LMS, no caso de se tratar de um SCO. Relembra-se que não existe qualquer processo comunicacional associado a um objecto de conteúdo do tipo *Asset*.
- **Login Session** – Período de tempo que vai desde o instante em que o aluno começa uma sessão (*logged on*), até ao momento em que a termina (*logged out*).

A figura 8 apresenta, de forma esquemática, o relacionamento temporal existente entre estas entidades.

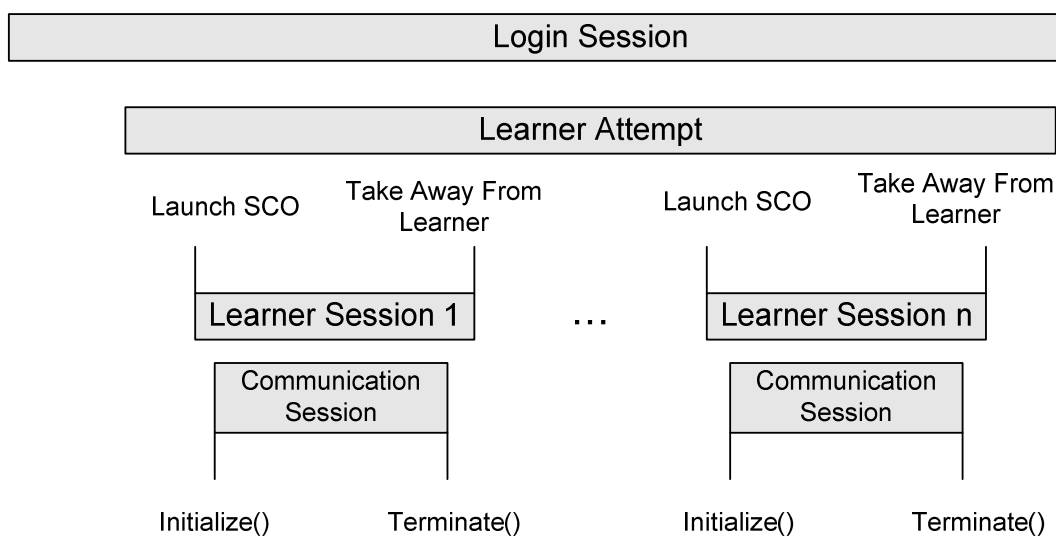


Figura 8 – Modelo temporal de relacionamentos envolvendo um SCO.  
Adaptado de SCORM Run-Time Environment Version 1.3 (ADL, 2004c).

Os conceitos que acabaram de ser descritos são importantes em termos da gestão que o RTE faz das interações com um SCO, sendo que, para um *Asset* o RTE apenas considera *Learner Attempts* e *Learner Sessions*.

Um *Learner Attempt* começa quando uma actividade tiver sido identificada como devendo ser iniciada. Durante esse processo, o aluno ficará envolvido com o CO (SCO ou Asset) associado à actividade.

Logo que o aluno esteja envolvido (CO lançado no *browser* do aluno), tem início uma *Learner Session*. Se o objecto de conteúdo lançado for um SCO, logo que este inicialize a comunicação com o LMS, tem início uma *Communication Session*. Esta terminará quando terminar a comunicação do SCO com o LMS.

Como acabou de se ver, as *Learner Sessions* podem deixar o SCO num estado “suspenso” (o aluno não cumpriu os pressupostos do SCO) ou num estado “normal” (o aluno terminou o *Learner Attempt* cumprindo todos os requisitos definidos pelo SCO).

Para um SCO, um *Learner Attempt* termina quando terminar num estado “normal” a sessão em que o aluno estava envolvido. Para um *Asset*, um *Learner Attempt* termina quando o aluno aceder com sucesso a esse recurso.

O último livro a referir é também o último conjunto de documentos gerado pelo projecto SCORM e só apareceu com a versão 1.3 ou 2004 do projecto.

Trata-se do SN (*Sequencing and Navigation Book*) (ADL, 2004d). Este livro descreve a forma como a informação sobre sequenciamento pode ser utilizada para definir as diversas estratégias de estabelecimento de sequências de apresentação das diversas componentes de um curso. Por outro lado, apresenta a forma como a informação de sequenciamento é interpretada durante o *runtime*, de forma a permitir a correcta tomada de decisão, quanto ao sequenciamento. Isto é, que item deverá seguir-se, de acordo com o percurso de aprendizagem seguido pelo aluno. Finalmente, apresenta também a forma como as solicitações de navegação, originadas pelas interações entre os utilizadores e os objectos de conteúdo, são processadas para identificar o próximo conteúdo a ser apresentado no *browser* do utilizador.

Partes deste novo livro são baseadas no *IMS Simple Sequencing Specification* do consórcio IMS. O termo *simple* apenas está relacionado com o limitado conjunto de situações de comportamento de sequenciamento descritas.

O sequenciamento, tal como é apresentado no *Sequencing and Navigation Book*, depende das seguintes componentes:

- Uma estrutura de actividades de aprendizagem definida - A “Árvore de Actividades” (*Activity Tree*);
- Uma estratégia de sequenciamento definida – O “Modelo de Definição de Sequenciamento” (*Sequencing Definition Model*);
- A aplicação de comportamentos definidos, como resposta a “eventos gatilho”, internos ou externos – Os “Comportamentos de Sequenciamento” SCORM (*SCORM Sequencing Behaviors*).

A figura 9 apresenta a relação directa, que no âmbito do projecto SCORM é possível estabelecer entre o conteúdo do “manifesto” (ficheiro XML com a organização dos conteúdos de um curso), e a “Árvore de Actividades” correspondente.



Um “manifesto” pode conter “submanifestos” e, desse modo, dar origem a uma “Árvore de Actividades” concordante como a que se apresenta na figura 10.

Qualquer elemento (<item>) “folha” de uma “Estrutura de Conteúdos” pode referenciar, quer um recurso (<resource>), quer um “Submanifesto” (<manifest>). Notar que um “Submanifesto” é ainda um “Manifesto”.

Dado que a informação de sequenciamento é aplicada em termos de actividades de aprendizagem estruturadas, uma estrutura de conteúdos bem formada fornece o ponto de partida necessário para a derivação de uma “Árvore de Actividades”.

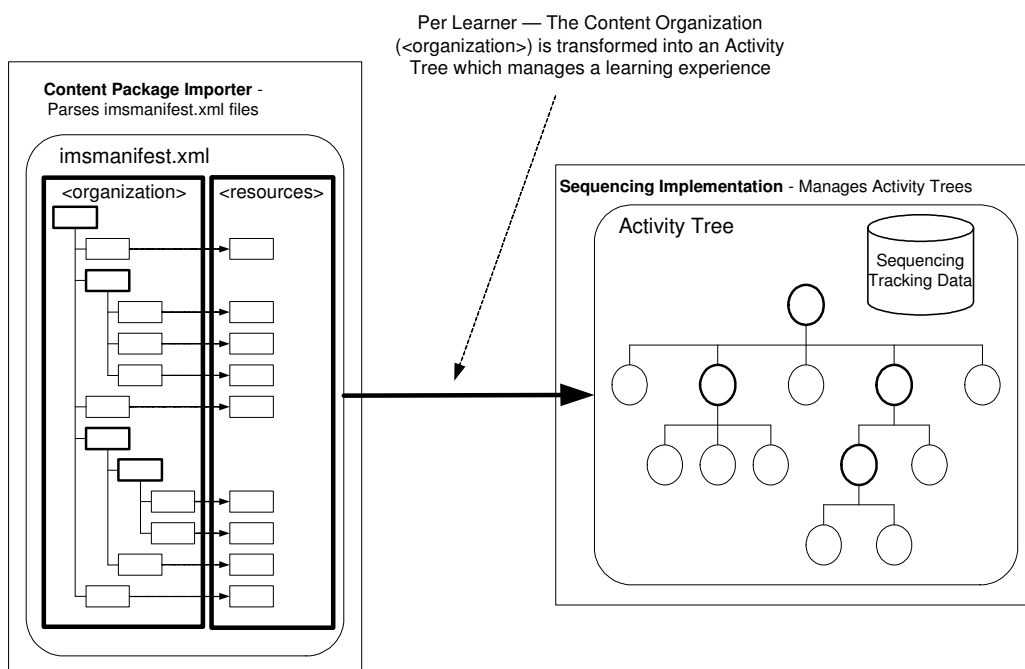


Figura 9 - Relação entre a Organização de Conteúdos (manifesto) e a Árvore de Actividades.  
Adaptado de (ADL, 2004d).

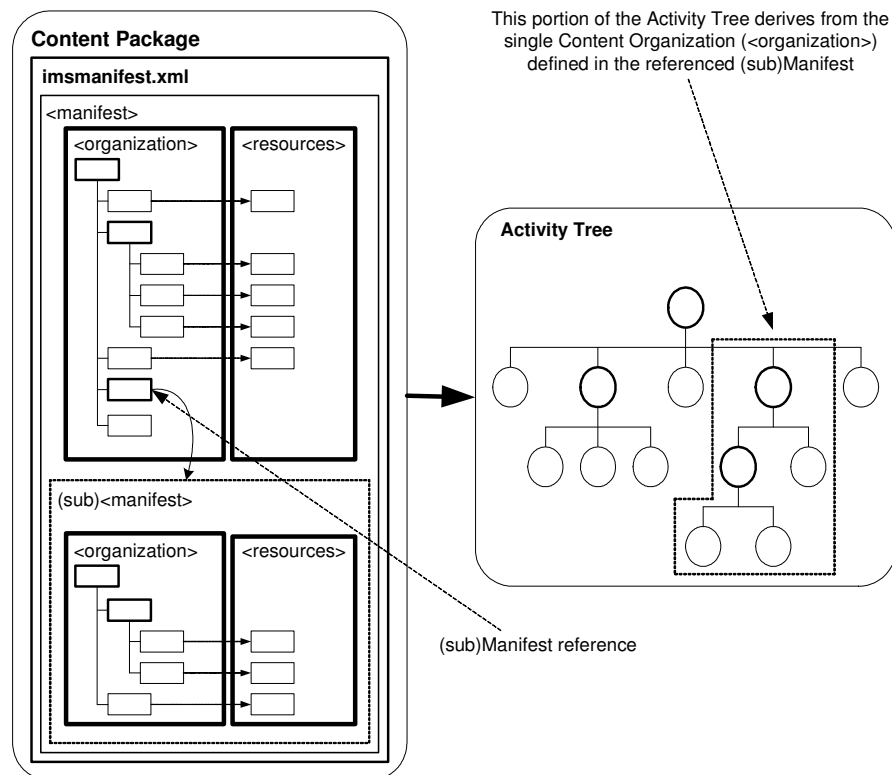


Figura 10 - Exemplo de uma Árvore de Actividades derivada a partir de um manifesto com submanifesto. Adaptado de (ADL, 2004d).

Para a estrutura que acabou de ser apresentada, o projecto SCORM utilizou mais uma componente do projecto IMS – o *IMS Simple Sequencing Specification*.

Um outro conceito importante na arquitectura proposta pelo projecto SCORM é o de *cluster* e a figura 11 apresenta um exemplo dessa entidade.

Um *cluster* corresponde à unidade elementar da “Árvore de Actividades”, em que os ramos do *cluster* são as subactividades da actividade raiz desse *cluster*. A actividade pai num *cluster* conterà a informação relativa à estratégia de sequenciamento das actividades do *cluster*. As actividades filhas (não *cluster*), terão associados *content objects* que serão seleccionados para apresentação, de acordo com a estratégia de sequenciamento definida.

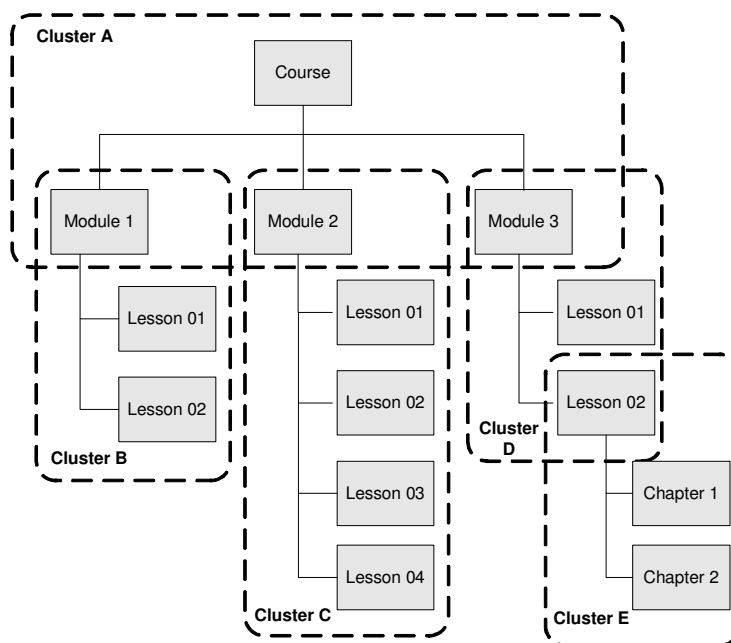


Figura 11 - Exemplo de Clusters. Adaptado de (ADL, 2004d).

Presentemente o *Sequencing and Navigation Book* não utiliza os metadados SCORM. Estes não têm qualquer impacto no processamento de pedidos de navegação ou comportamentos de sequenciamento.

Um *Content Package SCORM* pode representar um curso, uma aula, um módulo ou pode ser apenas uma colecção de CO relacionados, que devem ser armazenados num repositório SCORM.

No âmbito do projecto SCORM está previsto um único tipo de actividades – Actividades de Aprendizagem (*Learning Activities*).

Uma actividade de aprendizagem pode ser vista como uma unidade significativa de aprendizagem. Conceptualmente, algo que o aluno possa realizar durante o seu percurso de aprendizagem.

Uma actividade de aprendizagem pode fornecer um recurso de aprendizagem ao aluno ou pode ser composta por subactividades, Tais subactividades não se decompõem e são chamadas “folhas”, devendo possuir associadas a si, objectos de conteúdo (*Content Objects*).

Quando o LMS identifica uma actividade a disponibilizar ao aluno, lança o CO (*content object*) associado e apresenta-o no *browser* deste.

As LA (*Learning Activities*) são caracterizadas por:

- Possuir um “Início” e um “Fim”;
- Possuir a definição clara das condições de finalização;
- Poderem apresentar subactividades a vários níveis;
- Ocorrerem no contexto da “actividade pai”, caso esta exista.

## Capítulo 4

# Técnicas de captura e representação de conhecimento utilizadas

### 4.1 Introdução

É sabido que se pode encontrar no âmbito do desenvolvimento de sistemas de informação um vastíssimo leque de técnicas e metodologias de análise e de criação de modelos de sistemas, umas mais orientadas aos dados, outras que privilegiam os processos, outras ainda que permitem modelar o comportamento dos sistemas, etc.

A utilização combinada de diversas técnicas de modelação aumenta a probabilidade de uma melhor representação do sistema em análise.

Para tentar captar o máximo conhecimento acerca das interações dos diversos utilizadores com os LMS, utilizou-se um conjunto diversificado de técnicas de modelação/representação de conhecimento, tendo-se dedicado uma atenção especial à utilização da UML (Unified Modeling Language) (Booch et al, 1999a), (Booch et al, 1999b), (Fowler, 2004), (Eriksson, 1998), (Silva, 2001), (Nunes, 2001).

### 4.2 O modelo Entidades-Relacionamentos

Embora tratando-se de uma técnica já com muitos anos de existência e orientada apenas à representação dos dados e das relações que entre estes se estabelecem, constitui ainda hoje uma ferramenta de grande utilidade na captação e representação do conhecimento acerca dos dados que circulam, são tratados, armazenados e difundidos no âmbito de qualquer sistema ou organização.

O modelo E-R é um modelo conceptual que utiliza um conjunto reduzido de conceitos e elementos gráficos com base nos quais é possível o desenho dos dados e dos seus relacionamentos.

A figura 32 apresenta um excerto do modelo E-R obtido a partir das funcionalidades nativas do gestor de bases de dados Microsoft Access, com base no qual foram realizados alguns testes no âmbito do projecto ODL Toolbox.

### 4.3 Ontologias de domínio

Uma vez que em determinada fase da realização do trabalho que deu origem a esta tese se investiu bastante no estudo deste tipo de ontologias, o subcapítulo reflectirá esse esforço e tempo despendidos.

Dada a não estabilização deste domínio de conhecimento, é possível encontrar inúmeras definições sobre o que é uma ontologia e para que serve. Tais definições dependem muito do perfil e área de conhecimento a que está ligado quem as dá.

No contexto dos sistemas de informação, uma ontologia é definida como sendo um conjunto de conceitos e termos que podem ser utilizados para descrever uma área de conhecimento ou construir uma representação do conhecimento (Munoz, 2005).

Por outro lado, Noy (Noy, 2001a) afirma que uma ontologia é uma descrição formal e explícita de:

- Conceitos num dado domínio (há um certo paralelo entre conceito e classe);
- Propriedades de cada conceito ou classe, descrevendo diversas características e atributos do conceito ou classe (*slots* ou propriedades);
- Restrições aos *slots*.

Os autores Borst (Borst, 1997) e Gruber (Gruber, 1993) definem uma ontologia como sendo uma especificação formal de uma conceptualização partilhada.

Uma vez que uma ontologia é uma conceptualização, corresponde a um modelo abstracto de algum fenómeno do mundo real que identifica os conceitos relevantes associados a esse fenómeno.

Sendo uma especificação formal, uma ontologia é definida de forma a poder ser lida e interpretada por aplicações informáticas. Por fim, sendo uma conceptualização partilhada, uma ontologia captura conhecimento consensual, isto é, conhecimento aceite por um grupo de pessoas e não apenas por algum indivíduo em particular (Studer et al, 1998).

Existem diversos tipos de ontologias, agrupadas de acordo com as diferentes sensibilidades dos diferentes autores que as categorizam e de acordo com o critério base de classificação. Pode encontrar-se classificações quanto à função, quanto ao grau de formalismo, quanto à estrutura, quanto ao conteúdo, etc. (Almeida, 2003). Concretizando, é possível encontrar-se na bibliografia referências a ontologias gerais, ontologias linguísticas, ontologias de aplicação, ontologias de representação, ontologias de domínio, etc. (van Heijst et al. 1997; Mizoguchi et al. 1995).

Aquelas em que estamos interessados no âmbito deste trabalho, são as ontologias de domínio. Estas fornecem vocabulários sobre os conceitos e relações pertencentes a um domínio específico, sobre as actividades que aí têm lugar e sobre os princípios fundamentais que governam o domínio em análise.

De acordo com Noy (Noy, 2001), uma ontologia é o reflexo do mundo real num determinado domínio e se num caso particular não existir qualquer categorização, a ontologia correspondente deve reflectir tal realidade.

A mesma autora afirma que se utilizam ontologias para facilitar a comunicação entre peritos em determinado domínio e entre peritos num domínio e sistemas baseados em conhecimento. Então, será interessante reflectir na ontologia a visão que os peritos têm do domínio em causa.

Uma ontologia não deve conter toda a informação possível relacionada com o domínio. Não se deve especializar ou generalizar mais do que o necessário para a utilização que dela se quer fazer em determinado contexto.

São diversas as motivações e as áreas de aplicação das ontologias de domínio. Desde a medicina ao direito, passando pelos sistemas de informação geográficos, tem sido possível observar a sua aplicação em quase todas as áreas de conhecimento.

Em (Ramalho et al, 2003) é possível verificar que, por outro lado, as ontologias têm sido criadas para permitir obter diferentes “vistas” sobre a informação armazenada em bases de dados, documentos XML e outros recursos informacionais.

Segundo Noy (Noy, 2001a), são várias as justificações para a utilização de ontologias como forma de representar ou modelar um sistema:

- Para partilhar entendimentos comuns sobre a estrutura da informação, através de pessoas ou agentes de software;
- Para permitir a reutilização de conhecimento num determinado domínio;
- Para tornar explícitos determinados pressupostos de um domínio;
- Para separar conhecimento do domínio do conhecimento operacional;
- Para analisar conhecimento de um domínio.

Na prática, o desenvolvimento de uma ontologia implica:

- A definição das classes (conceitos) da ontologia;
- Estruturação das classes numa hierarquia taxonómica (subclasses-superclasses);
- Definição para cada classe, dos atributos (*slots*) e descrição dos valores permitidos para esses *slots*;
- Preenchimento dos valores dos *slots* para as instâncias das classes.

Tal como acontece com outras abordagens, também neste caso é aconselhável a enumeração de um conjunto de questões a colocar à ontologia (questões de competência) para perceber se a mesma contém ou não a informação necessária (Noy, 2001a).

Podem estabelecer-se comparações entre as questões de competência na engenharia de ontologias e os modelos de casos de uso no contexto da engenharia de software orientada a objectos (Guizzardi, 2005). Ambos os casos podem ajudar a concluir acerca da adequabilidade do modelo face à realidade representada.

Os conceitos e as relações entre eles formam a base das ontologias. Mas uma característica essencial das ontologias é a definição de axiomas. Os axiomas devem ser estabelecidos como forma de definir ou limitar a semântica dos termos. Os axiomas especificam definições de termos nas ontologias e restrições sobre a sua interpretação. A descrição dos axiomas pode e deve ser feita em linguagem natural, reflectindo simplesmente as restrições existentes sobre o universo de discurso (Guizzardi, 2005).



É possível identificar numa ontologia dois tipos de axiomas – axiomas de derivação e axiomas de consolidação (Falbo, 1998). Os primeiros permitem explicitar informações a partir de conhecimento previamente existente. Desta forma, tornam-se em meios de dedução e representam consequências lógicas neste processo. Por outro lado, os axiomas de consolidação são utilizados para descrever a coerência das informações existentes.

Falbo (Falbo, 1998) afirma que os axiomas de consolidação definem condicionantes para o estabelecimento de uma relação ou para a definição de um objecto como instância de um conceito.

Segundo Noy (Noy, 2001a), algumas das preocupações que devem estar presentes quando se constrói uma ontologia são:

- Garantir que a hierarquia de classes está correcta:
  - Verificar a transitividade de relações hierárquicas;
  - Prever a evolução da hierarquia de classes;
  - Distinguir entre uma classe e o seu nome;
  - Evitar classes e subclasses cíclicas.
- Analisar características de classes irmãs (mesmo nível hierárquico):
  - Que classes devem pertencer a um mesmo nível?;
  - Identificar o correcto número de classes, acima do qual se terá um número exagerado (difícil de manipular) ou abaixo do qual deixará de ser possível uma representação capaz no universo de discurso.
- Determinar heranças múltiplas;
- Quando introduzir ou não uma nova classe?;
- Decidir se incluir uma nova classe ou se uma nova propriedade de uma classe já existente;
- Decidir se é necessária uma nova classe ou se uma instância de uma classe já existente;
- Identificar classes disjuntas;
- Identificar *slots* inversas (Ex: A é produzido por B; B produz A);
- Identificar valores por defeito de determinadas *slots*;
- Definir uma regra de atribuição de nomes às classes, às *slots* e às instâncias.

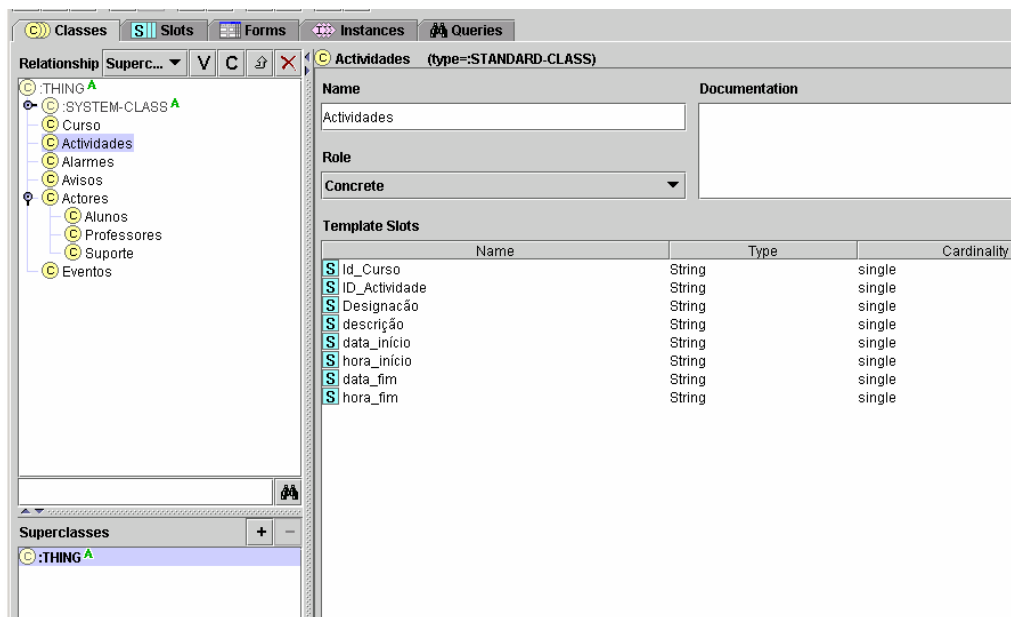


Figura 12 – Vista parcial da ontologia relativa à gestão de cursos na Web (obtida através da ferramenta Protégé2000).

A figura 12 apresenta um excerto da ontologia de domínio criada para a definição da camada de gestão operacional dos cursos na Web, com recurso ao editor de ontologias Protégé2000<sup>45</sup>.

De realçar, como seria de esperar, as semelhanças existentes entre este modelo (ontologia) e o modelo E-R da figura 32, no que toca às classes da ontologia e às entidades do modelo E-R.

Em termos de geração de metadados, o Protégé2000 dá uma ajuda importante ao permitir obter essa informação em diversos formatos.

Apresenta-se na figura 13 um excerto dos metadados obtidos automaticamente com base na ontologia da figura 12, relativos à classe “Actores” do modelo referido.

<sup>45</sup> <http://protege.stanford.edu/>

**Project: teste1**  
 Class Actores  
 Concrete Class Extends  
 :THING|  
 Direct Instances:  
 None  
 Direct Subclasses:  
 1. [Alunos](#)  
 2. [Professores](#)  
 3. [Suporte](#)

Template Slots					
Slot name	Documentation	Type	Allowed Values/Classes	Cardinality	Default
<u>Password</u>		String		0:1	
<u>id_actor</u>		String		0:1	
<u>Username</u>		String		0:1	
<u>e-mail</u>		String		0:1	
<u>id_curso</u>		String		0:1	

Figura 13 – Metadados referentes à ontologia de gestão de cursos na Web (Obtida através da ferramenta Protégé2000).

A ferramenta permite ainda gerar ficheiros em formato XMI (*XML Metadata Interchange*) correspondentes às ontologias criadas, capazes de serem lidos por variadas ferramentas, e de permitir que de uma forma automática, tais ferramentas possam gerar outros formatos de representação da mesma informação, por exemplo, os correspondentes diagramas de classes UML.

Existem diversas abordagens para a modelação de sistemas, muitas delas apresentando características muito semelhantes. Como se viu acima, não há, uma grande diferença entre as entidades de um modelo E-R e as classes ou conceitos utilizados numa ontologia.

Da mesma forma se não verificam grandes diferenças entre os atributos num modelo E-R e as *slots* utilizadas na caracterização de uma classe de uma ontologia.

Pensando, contudo, no que seria o diagrama E-R relativo ao problema em análise neste texto, não seria possível concluir a partir da semântica associada a esse tipo de diagramas, que, dado um curso específico, um actor não pode ser simultaneamente do tipo “professor” e “aluno”. Apenas questões de bom senso permitiriam deduzir essa regra.

Ao construir a ontologia da Figura 12, poder-se-ia associar-lhe um conjunto de axiomas que ajudassem a estabelecer inequivocamente essa semântica.

Um axioma pode ser uma frase em lógica de primeira ordem que, por definição, não necessita de prova ou demonstração.

Entre os filósofos gregos antigos, um axioma era uma reivindicação que poderia ser vista como verdadeira sem nenhuma necessidade de prova.

Na epistemologia, um axioma é uma verdade auto-evidente, na qual outros conhecimentos se devem apoiar e a partir da qual outro conhecimento é construído.

Um axioma, tal como é usado na Matemática moderna, significa um ponto de partida num sistema lógico<sup>46</sup>.

No caso concreto das ontologias, e do Protégé<sup>2000</sup> em particular, em vez de axiomas utilizam-se restrições para obter o mesmo efeito. Assim, retomando a questão referida atrás, quanto à impossibilidade de um actor ser simultaneamente aluno e professor, ter-se-ia que implementar uma restrição equivalente ao axioma em lógica de primeira ordem que se apresenta a seguir:

$$(\forall a,b) ( \text{professor}(a) \wedge \text{aluno}(b) \rightarrow \text{diferente}(a,b) )$$

A leitura a fazer deste axioma é a seguinte: Dados dois quaisquer actores a e b, se a é professor e b é aluno, então a tem que ser diferente de b.

## 4.4 A linguagem UML

A UML, criada por Peter Coad, Grady Booch, Ivar Jacobson e Jim Rumbaugh (Booch et al, 1999a) e (Booch et al, 1999b), é uma família de notações gráficas que ajuda na descrição e na modelação de software, principalmente quando se tratar de software orientado a objectos (Fowler, 2005).

Trata-se de uma norma relativamente aberta, controlada pelo OMG (*Object Management Group*)<sup>47</sup>, constituindo um consórcio aberto de diversas organizações/empresas.

Apesar de ser apresentada como uma norma, a verdade é que não se encontra uma forma única para a sua utilização.

---

<sup>46</sup> Estas e outras asserções sobre axiomas encontram-se disponíveis em: <http://pt.wikipedia.org/wiki/Axioma>.

<sup>47</sup> <http://www.omg.org/>

Fowler (Fowler, 2004), e Stephen Mellor (Mellor, 2005), identificaram três contextos de utilização da UML: *Sketch* (esboço), *Blueprint* (documento técnico de especificação) e “*Programming Languages*”.

Segundo Fowler (Fowler, 2004), na maior parte dos casos, os utilizadores usam a UML como *Sketch* para captar e comunicar aspectos particulares dos sistemas.

A figura 14 representa o conjunto de todos os tipos de diagramas que constituem a UML, tendo sido seleccionados para utilização neste trabalho, os Diagramas de Caso de Uso e os Diagramas de Sequência, como forma de realizar a análise de requisitos associada ao problema subjacente à escrita desta tese.

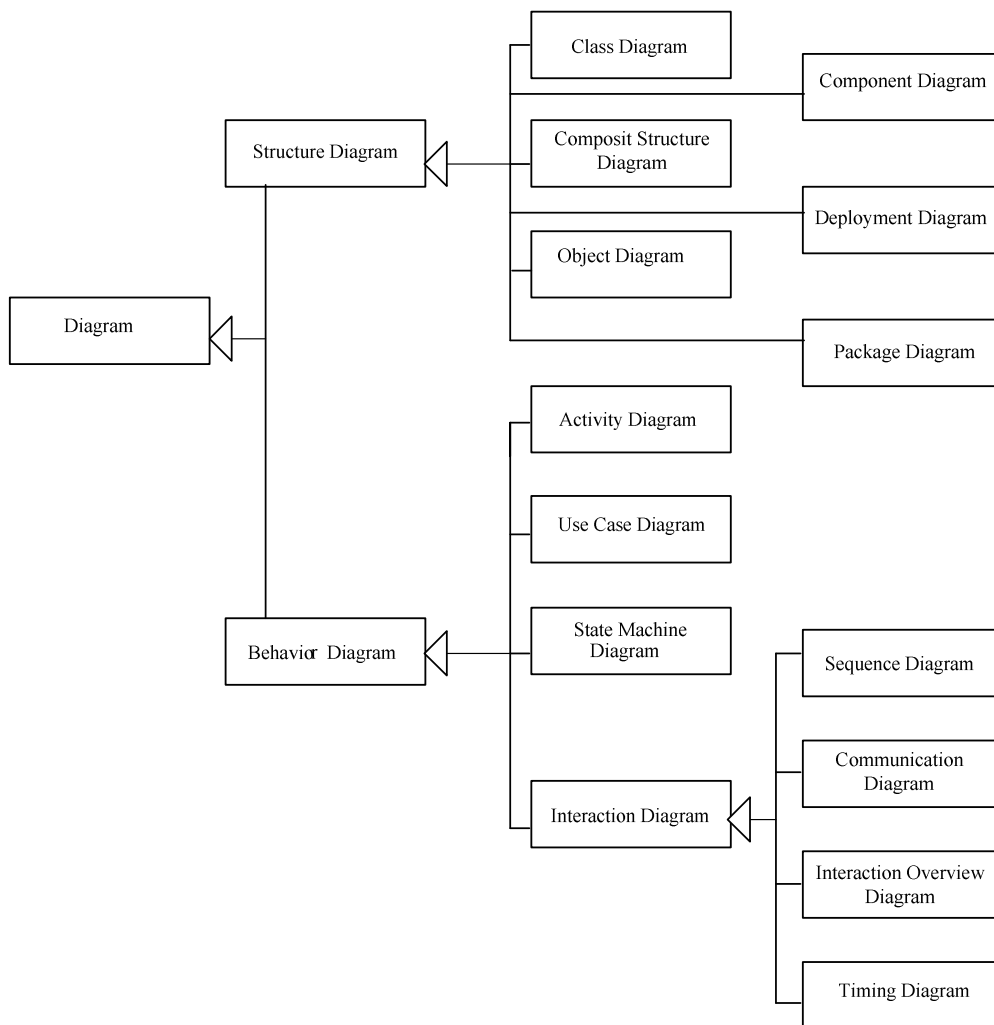


Figura 14 – Classificação dos tipos de diagramas UML.  
Adaptado de UML Distilled Third Edition (Fowler, 2004).

Os Diagramas de Caso de Uso são uma técnica para capturar os requisitos funcionais de sistemas, tendo sido utilizados para descrever as interações típicas que os diversos tipos de utilizadores estabelecem com os LMS. Quando falamos no âmbito de casos de uso (Use Cases), um Actor deve ser visto como uma “função” (*role*) que um utilizador desempenha na sua interação com o sistema.

Cada caso de uso tem um actor principal que “invoca” o sistema, solicitando o fornecimento de um serviço. Embora tal possa não acontecer, normalmente é o actor principal que dá início ao caso de uso.

Um caso de uso pode utilizar uma forma gráfica ou uma forma textual. No primeiro caso ter-se-á um Diagrama de Caso de Uso, enquanto que no segundo ficará disponível um documento mais preciso (rico), no que toca à descrição das interações com o caso de uso.

Um Diagrama de Casos de Uso de nível zero pode ser visto como um Diagrama de Contexto nos Métodos Estruturados, tal como acontece na Análise Estruturada (deMarco, 1978).

É importante lembrar que os casos de uso representam uma visão externa do sistema e, portanto, não é de esperar qualquer relação entre estes e as classes que virão a ser identificadas e utilizadas dentro do sistema.

Cada etapa de um caso de uso deve ser uma tarefa simples e deve mostrar claramente quem beneficia com essa etapa. Deve mostrar a intenção do actor e não a forma como ele a materializa.

Para além dos Diagramas de Casos de Uso, foram também utilizados Diagramas de Sequência. Quanto à sua finalidade, deve ter-se presente que os mesmos podem e devem ser usados quando se pretende observar o comportamento de diferentes objectos dentro de um mesmo caso de uso.

A linguagem UML aparece como resultado da evolução dos processos de desenvolvimento de software, com o surgimento do paradigma orientado a objectos e, por esse motivo, muitas técnicas que dela fazem parte foram desenvolvidas pensando em termos dos objectos, seu estado e comportamento. No caso concreto dos Diagramas de Sequência, originalmente eles servem para representar as interações entre os objectos relevantes do sistema em desenvolvimento. Contudo, em estados iniciais do processo de desenvolvimento de sistemas de informação, não é espectável que tais objectos estejam já

identificados. A técnica de modelação pode contudo utilizar-se com o objectivo de capturar e validar o máximo conhecimento acerca do sistema em análise.

Assim, adoptando uma utilização proposta em (Machado, 2005), apresentar-se-á na secção 5.1.2 um conjunto de Diagramas de Sequência, em que actores e casos de uso substituem os esperados objectos, fazendo-se a representação das interacções entre eles e definindo cenários passíveis de serem encontrados no contexto do trabalho realizado.

## Capítulo 5

# Estudo, proposta e discussão de uma camada de gestão de actividades em tempo real

### 5.1 Captura de conhecimento e modelação do sistema

#### 5.1.1 Introdução

Ao iniciar-se esta secção, deve ser realçado o facto de as preocupações de gestão que estiveram subjacentes a este trabalho estarem relacionadas com os aspectos operacionais dos cursos a distância, e não com a correspondente gestão administrativa. Isto é, as preocupações centraram-se na necessidade de gerir as interações dos utilizadores com as plataformas (LMS) e os cursos, face a um planeamento de actividades previamente estabelecido.

Num primeiro momento foi levada a cabo a identificação das diferentes fases que estão presentes na arquitectura e exploração de um curso a distância, com realce para a caracterização das diversas actividades de ensino/aprendizagem normalmente existentes nesse contexto, observáveis e, portanto, passíveis de serem geridas (Ramos et al, 2001).

Contudo, desde o início que se tornou clara a necessidade de um nível mais abrangente de caracterização do problema, como forma de melhor compreender e tratar a situação de maior detalhe, que era a que interessava estudar no âmbito deste trabalho. Isto é, a gestão em tempo real da execução das actividades de ensino/aprendizagem mediadas por LMS.



Nesta fase do trabalho, apesar de ter sido escolhida a linguagem UML como fornecedora principal das técnicas para representar o conhecimento capturado acerca do sistema, apenas se utilizou um subconjunto de todas as que a linguagem disponibiliza. Em concreto, para o levantamento de requisitos funcionais, foram utilizados Diagramas de Caso de Uso, e para representação do comportamento do sistema em determinadas situações (cenários), fez-se apelo a Diagramas de Sequência estereotipados (Machado, 2005), já que estes últimos consistem numa utilização adaptada da técnica original proposta por Booch (Booch et al, 1999a), (Booch et al, 1999b).

Não teria sido difícil descrever as interacções entre os diversos utilizadores e um LMS, sem que fosse necessário recorrer às referidas técnicas de representação gráfica e criação de modelos. Contudo, a opção pela sua utilização encontra justificação no facto de se tratar de uma linguagem normalizada e de reconhecimento universal. Como tal, um grande número de pessoas conseguirá apreender a informação que os diagramas transportam, sem necessidade de qualquer leitura complementar.

### 5.1.2 Usando UML para modelar as interacções com um LMS

Antes de se apresentarem os diagramas de caso de uso achados mais relevantes, será conveniente fazer uma chamada de atenção para o facto de estes poderem ser obtidos de acordo com uma filosofia orientada ao sistema ou orientada aos actores, isto é, tendo como referência os actores externos ou o sistema. Não raras vezes se observa na própria literatura existente, uma errada utilização desta técnica de modelação, aparecendo questões conceptuais como esta, ignoradas ou tratadas de forma incorrecta.

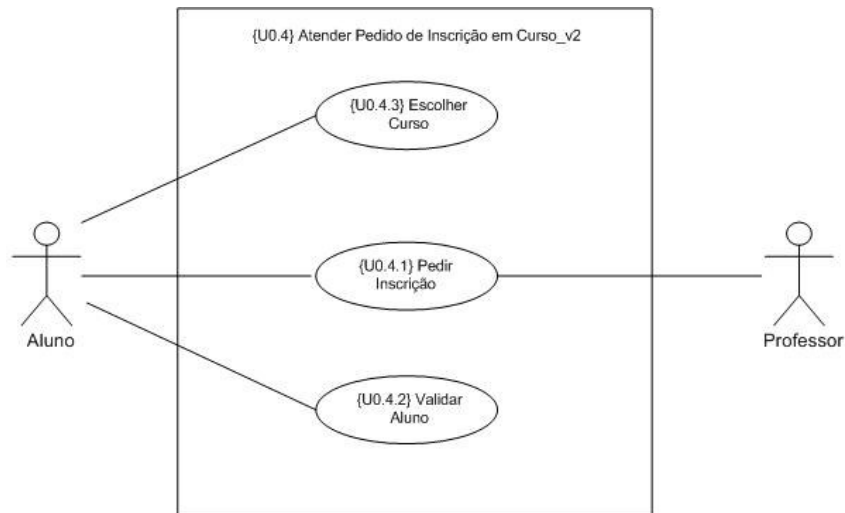


Figura 15 – Diagrama de Caso de Uso “Atender Pedido de Inscrição em Curso\_v2”.

A título de exemplo, imagine-se o diagrama de caso de uso relativo à actividade de inscrever um aluno num dado curso, por solicitação deste. A figura 15 mostra como o tipo de erro referido anteriormente pode ocorrer com a maior das facilidades.

No Diagrama da figura 15 é possível perceber que existem dois tipos de actores que interagem com o sistema. Este é delimitado pela fronteira que separa os casos de uso que constituem o diagrama, dos actores “Aluno” e “Professor”.

Parece não haver dúvidas quanto à intenção de cada um dos actores. O “Aluno”, depois de validado pelo sistema, escolhe o curso em que quer participar e solicita a sua inscrição nesse curso.

Por seu lado, o “Professor”, avalia todos os pedidos de inscrição que tenham sido efectuados, inscrevendo os “Alunos” de acordo com algum critério. Repare-se no entanto, com mais cuidado, nos casos de uso do diagrama. Dois deles representam nitidamente actividades da autoria do actor “Aluno”. Trata-se dos casos de uso “Escolher Curso” e “Pedir Inscrição”. Contudo, o terceiro caso de uso, “Validar Aluno” consiste numa actividade do sistema e não de qualquer dos actores!

Uma simples mudança nos identificadores dos casos de uso permitiria tornar consistente o diagrama da figura 15, havendo contudo que decidir onde se pretende colocar a ênfase do funcionamento do sistema. Nos actores ou no sistema propriamente dito.

Deduz-se do que se acabou de dizer, que uma correcta escolha dos identificadores dos casos de uso é, nesta fase, de importância vital para a semântica dos diagramas.

Apresentam-se de seguida os Diagramas de Casos de Uso e os Diagramas de Sequência mais relevantes referentes à análise e modelação das interacções dos utilizadores com um LMS. A figura 16 apresenta a visão de mais alto nível, isto é, o nível zero dessas interacções.

O diagrama mostra a existência de um conjunto de actores que interagem com o LMS e chama-se a atenção para quatro aspectos achados relevantes. O primeiro tem que ver com o actor “Actor”. A designação para este tipo de actor, embora pouco aconselhável pela confusão que pode originar, é muito anterior à decisão de utilizar a UML para a modelação do sistema, e refere aquele conjunto de entidades que participam activamente no funcionamento de um curso (Alunos, Professores e Elementos das Equipas de Suporte). Em segundo lugar, a representação do estereótipo “Relógio”, interagindo com o caso de uso “Emitir Notificações”. Como se verá mais adiante neste documento, esta actividade deve ser executada de forma automática e repetida indefinidamente, não havendo outro “gatilho” para a iniciar que não seja o passar do tempo. Daí a representação efectuada. O terceiro aspecto tem que ver com a divisão do conjunto dos actores em três grupos principais. O grupo daqueles que participam activamente no curso (“Actor”), os que de alguma forma se relacionam com o LMS mas de uma forma passiva, isto é, é o LMS que toma a iniciativa de os envolver ou invocar, nomeadamente através do envio de notificações, e um último grupo, constituído apenas por um tipo de actor, o “Administrador do LMS”.

No segundo grupo cabem os “Responsáveis” e os “Supervisores” e o último tem a sua justificação no facto de intervir activamente ao nível do registo ou criação de contas de novos utilizadores do LMS, não participando contudo nos cursos propriamente ditos.

O quarto aspecto a que deve ser dedicada alguma atenção, prende-se com a representação do conceito de especialização/generalização na representação que é feita dos actores “Aluno”, “Professor” e “Suporte” relativamente ao actor “Actor”.

Em alguns casos, optou-se pela repetição da representação dos actores, nomeadamente no caso do “Aluno” e do “Professor”. Esta repetição justifica-se pelo facto de, por um lado permitir uma maior clareza na leitura do diagrama, ao evitar o cruzamento de linhas de relacionamento, e por outro, separar o que são interacções comuns a qualquer “Actor” (Aluno, Professor ou Suporte), daquelas que dizem respeito apenas a algum dos actores individualmente.

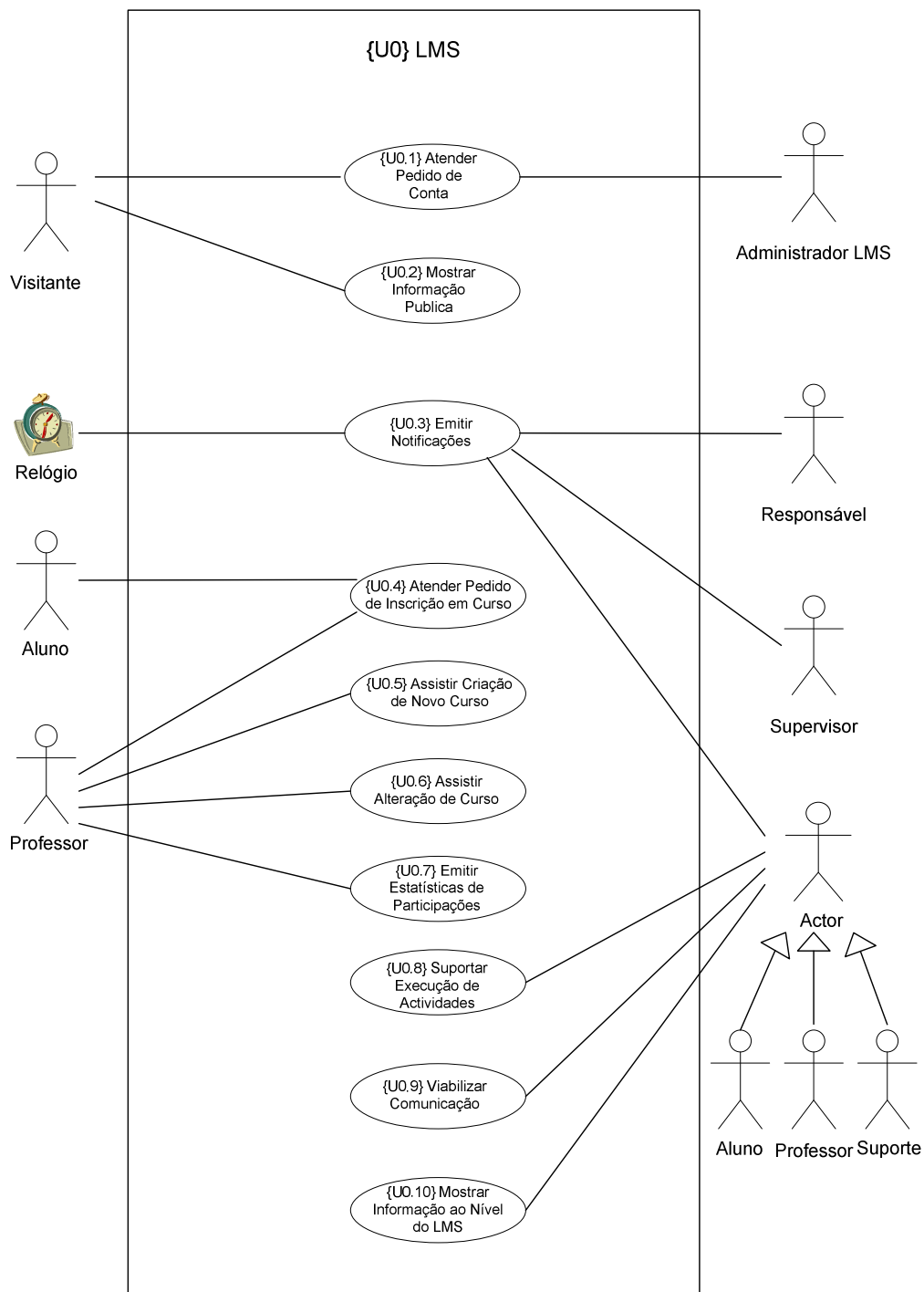


Figura 16 – Diagrama de Caso de Uso “{U0}LMS” (Nível Zero).

O detalhe dos casos de uso do diagrama de nível zero será mostrado de seguida, utilizando os principais diagramas de casos de uso e derivando, para cada um deles, o correspondente Diagrama de Sequência.

Será conveniente realçar, neste momento, o facto de terem sido utilizadas algumas das novas funcionalidades da UML, tornadas possíveis pela versão 2.0 da linguagem, nomeadamente a nível dos diagramas de sequência (Fowler, 2004).

Como exemplos dessas novas funcionalidades, a possibilidade de representação num mesmo diagrama de sequência, de cenários alternativos e de execução cíclica de actividades.

Pela sua maior relevância para o entendimento do funcionamento das interacções dos utilizadores com os LMS, descrever-se-ão os casos de uso seguintes:

- {U0.1} Atender Pedido de Conta;
- {U0.3} Emitir Notificações;
- {U0.4} Atender Pedido de Inscrição em Curso;
- {U0.5} Assistir Criação de Novo Curso;
- {U0.8} Suportar Execução de Actividades

### **{U0.1} Atender Pedido de Conta**

Este caso de uso representa os relacionamentos que se podem encontrar quando se está em presença de uma situação em que um “Visitante” (utilizador não registado no sistema) solicita o seu registo. A esta solicitação corresponderá, se aceite, a criação de uma conta com um *username* e uma *password*.

A figura 17 apresenta o diagrama de casos de uso correspondente a este tipo de interacção.

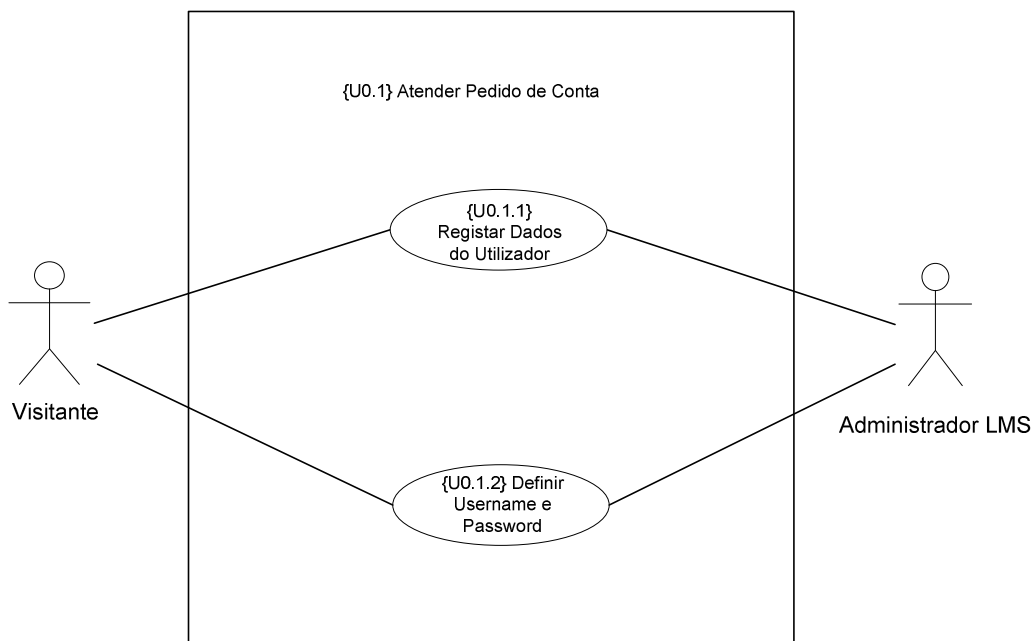


Figura 17 - Diagrama de Caso de Uso {U0.1} Atender Pedido de Conta.

Caso se pretenda saber como são levadas a cabo essas tarefas, ter-se-á que apelar para o diagrama de sequência correspondente. Tal diagrama é apresentado na figura 18.

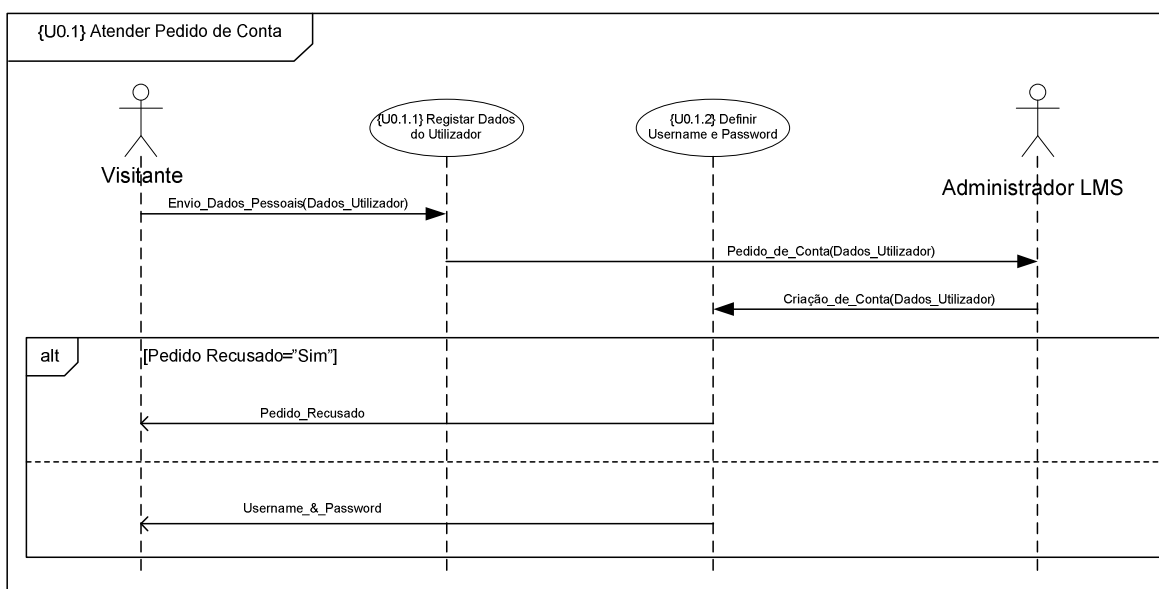


Figura 18 - Diagrama de Sequência do Diagrama de Casos de Uso {U0.1} Atender Pedido de Conta.

Neste tipo de diagramas, o eixo vertical está associado ao tempo, devendo a evolução do mesmo ser lida de cima para baixo.

A cada actor ou caso de uso (objectos na versão original) está associada uma linha a tracejado, designada por Linha de Vida do Objecto. Numa utilização tradicional destes diagramas, sobre essa linha deve ser representado o objecto, desde o instante do seu aparecimento no processo, até ao momento em que deixa de existir.

Como se explicou, foi feita uma utilização adaptada destes diagramas, com a intenção de proceder ao levantamento de requisitos do sistema e, assim, em vez de objectos temos casos de uso e actores. Por esse motivo, não se encontram representados sobre essas Linhas de Vida, quaisquer objectos. Esta utilização é aceitável se for considerado que numa fase tão inicial do processo, não existe ainda qualquer indicação clara relativamente aos objectos que serão necessários aquando do desenvolvimento do software.

Na figura 18, todos os actores e casos de uso têm uma linha de vida com a mesma duração, já que o sistema (os casos de uso associados ao LMS) e os actores que com ele interagem, existem desde sempre e enquanto a utilização do LMS pelos utilizadores se verificar.

Em todo o caso o mais importante nestes diagramas é a ideia clara de sequência que é transmitida ao seguir-se as diversas mensagens que vão sendo geradas e enviadas a cada elemento dos diagramas.

Para um aprofundamento do conhecimento sobre esta linguagem, seus componentes (diagramas) e sua utilização, sugere-se a consulta de (Booch et al, 1999a), (Booch et al, 1999b), (Nunes, 2001), (Eriksson, 1998), (Silva, 2001) e (Fowler, 2004).

O que é possível observar no diagrama de sequência da figura 18 é que o actor “Visitante” dá início ao processo, enviando uma mensagem “Envio\_de\_Dados\_Pessoais(Dados\_pessoais)” ao caso de uso “{U0.1.1} Registrar dados do Utilizador”. Este caso de uso faz a recolha desses dados e após o seu registo provisório, envia-os ao actor “Administrador LMS”. Com base em algum critério, o administrador do LMS dá posterior seguimento ao processo, reenviando os dados do pretendente, ao caso de uso “{U0.1.2} Definir *Username e Password*” que, tendo em conta um conjunto de testes e validações, gera um novo *username* e uma nova *password*. De alguma forma estes dados chegarão ao novo utilizador por meio da mensagem “Envio\_Dados\_Conta(*Username, Password*), caso o pedido tenha sido aceite. No caso em que esse pedido tenha sido

recusado, o “Visitante” receberá uma notificação “Pedido\_recusado”, alertando-o para esse facto.

Em bom rigor, devia incluir-se no diagrama de casos de uso da figura 17 um terceiro actor – o “Utilizador”. É que, quando solicita o pedido de criação de uma nova conta, o actor é apenas um “Visitante”, mas quando o seu pedido é aceite e lhe é comunicado o seu *username* e a sua *password*, ele já é, de facto, um novo “Utilizador” do sistema.

Atente-se seguidamente no Diagrama de Casos de Uso “{U0.3} Emitir Notificações” apresentado na figura 19. Este diagrama retrata a funcionalidade que se propõe no trabalho presente, relativa à emissão de notificações destinadas a alertar os diversos actores para situações de não execução de actividades planeadas, dentro das janelas temporais previstas.

Como se referiu anteriormente, foi incluído o estereótipo “Relógio” com a intenção de representar o funcionamento automático do sistema no que toca à actividade de detecção de situações justificadoras de notificações. Esta representação poderia ser substituída pela ausência de qualquer actor ou estereótipo a nível do diagrama de casos de uso e pela inclusão de uma mensagem com origem “no nada” (*found message*), responsável por dar início à actividade de teste da existência de situações motivadoras de notificações (Fowler, 2004).

O diagrama de sequência correspondente é apresentado na figura 20 e apresenta uma particularidade que não tinha sido referida até ao momento. Salvaguardadas as devidas diferenças, o bloco representando alternativas (alt) pode ser visto como uma estrutura CASE ou um conjunto de estruturas IF aninhadas, tal como são utilizadas em programação estruturada.

A mensagem “Informa\_Notificação(Notificação\_ID, Tipo, Destino\_ID)”, recebida do caso de uso “{U0.3.2} Detectar Situação de Notificação”, permite ao caso de uso “{U0.3.1} Enviar Notificação”, desencadear o processo de envio de uma notificação (“Aviso”, “Primeiro alarme”, “Segundo alarme” ou “Informação”) ao destinatário (“Actor”, “Supervisor” ou “Responsável”) identificado na mensagem pelo argumento “Destino\_Id”.



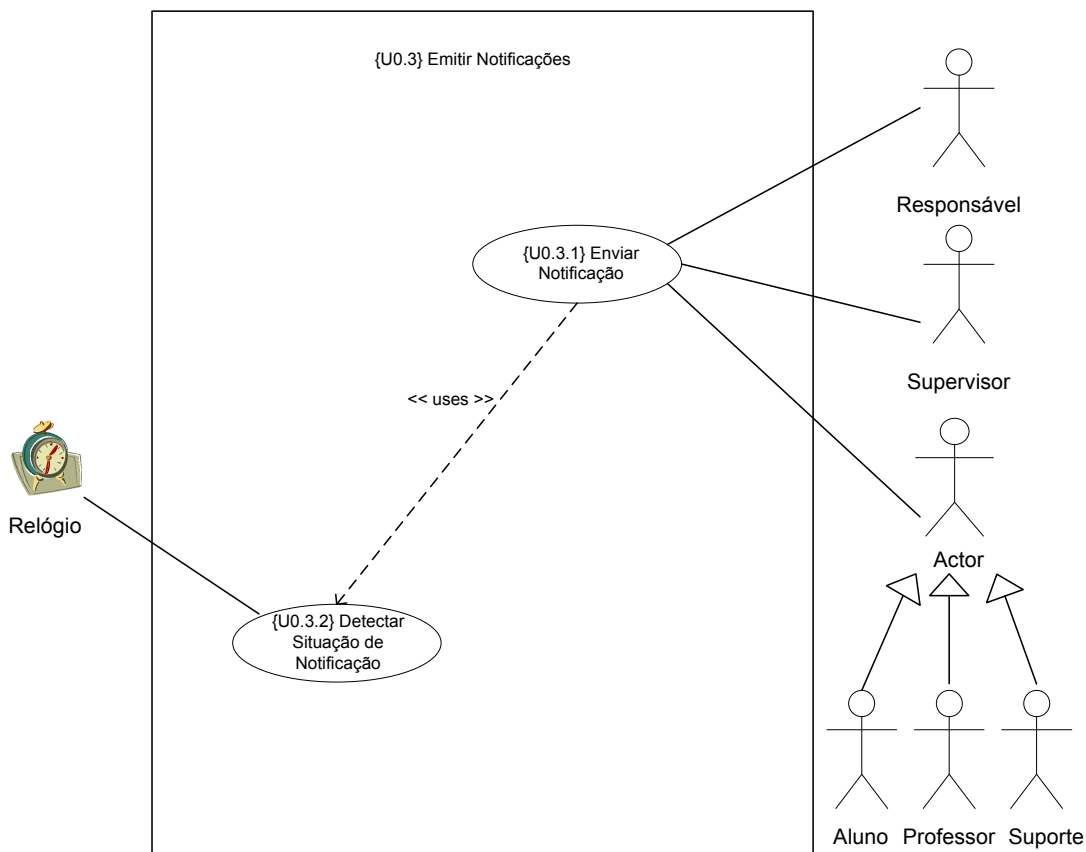


Figura 19 - Diagrama de Caso de Uso {U0.3}Emitir Notificações.

Nesta notação, cada linha tracejada horizontal separa uma situação alternativa da seguinte, sendo que para o bloco “alt”, em cada momento, apenas uma das alternativas será executada, dependendo do resultado do teste condicional efectuado.

De realçar ainda no diagrama da figura 20 a inclusão de uma outra estrutura, disponível a partir do lançamento da UML 2.0. Trata-se do bloco *loop*, que permite representar a repetição de interações dentro de um “Diagrama de Sequência. Esta representação corresponde ao que em programação estruturada poderia ser a utilização de uma estrutura “REPETIR <acção> ATÉ <condição>” ou, “ENQUANTO <condição> FAZER <acção>”.

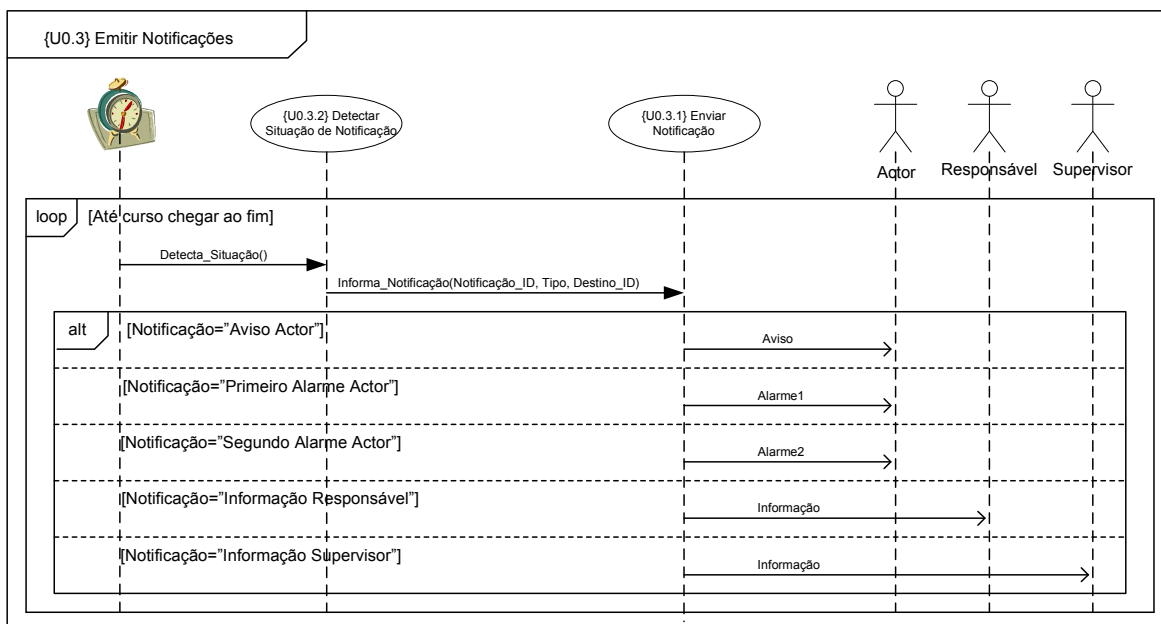


Figura 20 - Diagrama de Sequência do Diagrama de Casos de Uso {U0.3} Emitir Notificações.

#### {U0.4} Atender Pedido de Inscrição em Curso

Uma actividade importante quando se pensa em termos de frequência de cursos suportados por LMS consiste na inscrição prévia dos utilizadores (alunos) no curso desejado. A figura 21 representa o diagrama de casos de uso correspondente a essa funcionalidade e nele é possível ver que a interacção do utilizador (“Aluno”) com o sistema (LMS) se faz ao nível do caso de uso “Inscrever em Curso”. Este, por sua vez, invoca as funcionalidades de um conjunto de outros casos de uso (invocações essas representadas no diagrama por relações do tipo <<uses>>) que permitirão validar o utilizador como podendo ou não ser inscrito no curso, mostrar ao utilizador os cursos disponíveis, aceitar uma eventual escolha por parte deste e decidir sobre a realização da inscrição.

Uma solicitação do LMS com o objectivo descrito, provocará a aplicação automática de um conjunto de regras de avaliação da possibilidade de o aluno poder ser inscrito ou não no curso em causa, sendo o resultado de tal avaliação comunicado quer ao aluno, quer ao professor.

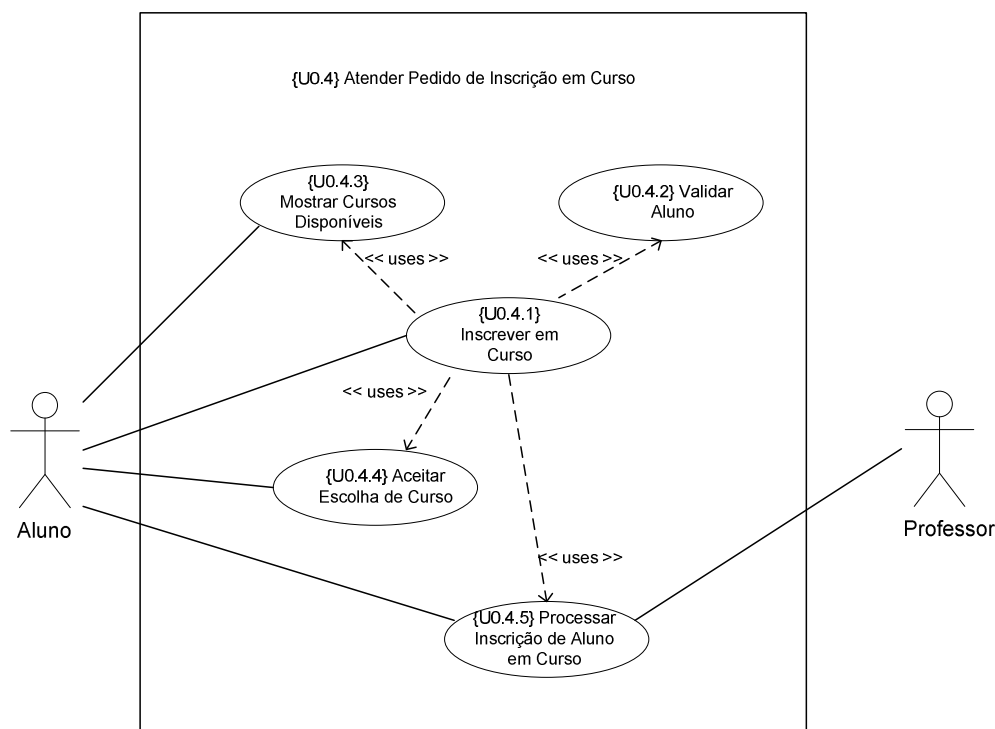
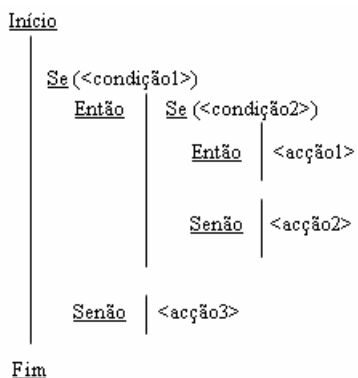


Figura 21 - Diagrama de Caso de Uso {U0.4} Atender Pedido de Inscrição em Curso.

O correspondente Diagrama de Sequência é apresentado na figura 22, motivando algumas considerações adicionais relativas ao seu desenho. De notar que este mecanismo funciona como tratando-se da condição utilizada numa estrutura IF <condição> THEN <acção1> ELSE <acção2>, comum na programação estruturada.

Por outro lado, este é um segundo bloco “alt” aninhado dentro do primeiro. O diagrama de sequência representa, deste modo, um comportamento semelhante ao que seria imposto por um bloco algorítmico do tipo:



Como se pode constatar da observação do diagrama de sequência da figura 22, o pedido de inscrição no curso pode ser rejeitado por o aluno não reunir as condições necessárias, ou por haver um qualquer tipo de impedimento ao nível do curso. Por exemplo, já estar preenchido com o número máximo de alunos previsto para o mesmo.

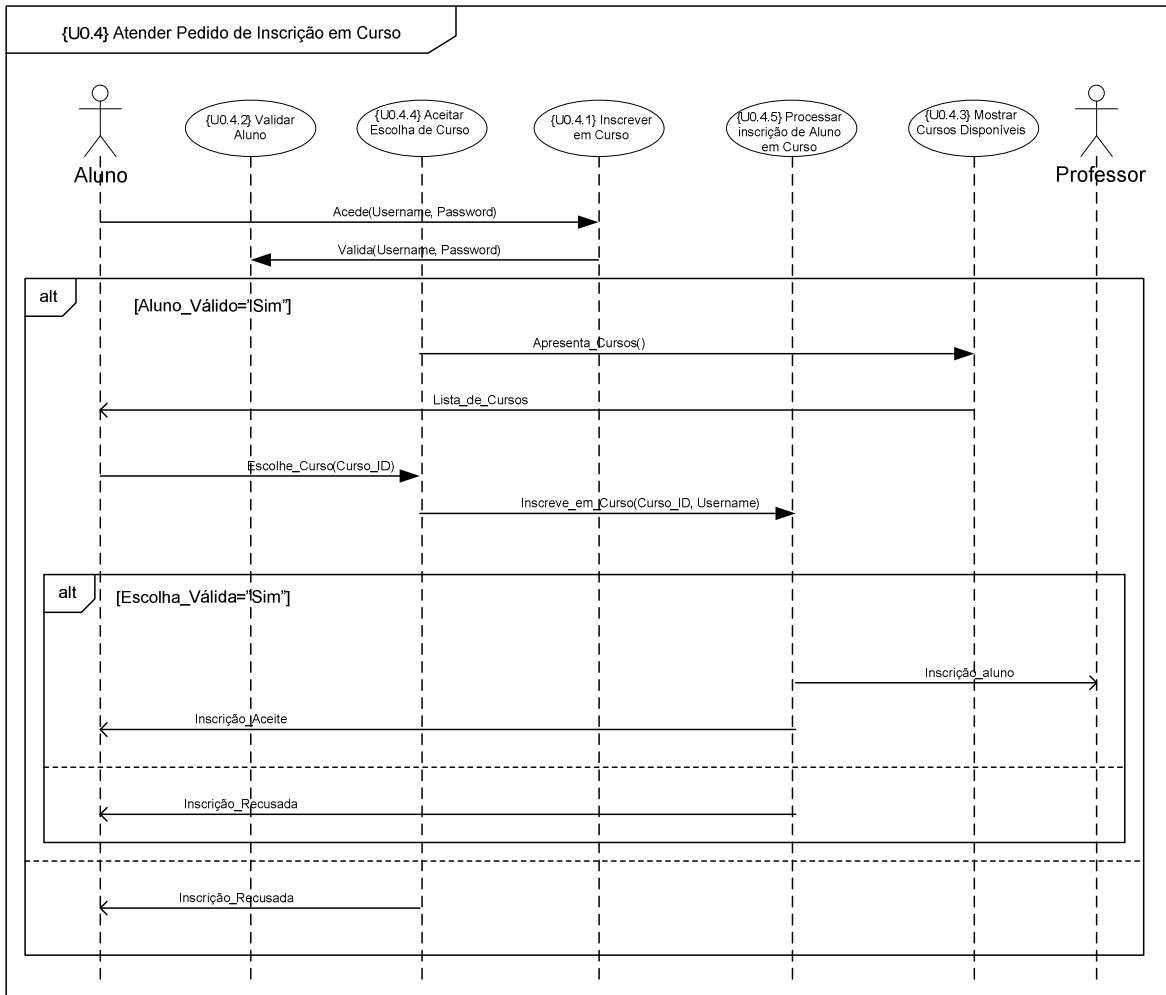


Figura 22 - Diagrama de Sequência do Diagrama de Casos de Uso {U0.4} Atender Pedido de Inscrição em Curso.

### {U0.5} Assistir Criação de Novo Curso

A actividade de criação de um curso é normalmente da responsabilidade do actor “Professor” e quando ocorre a solicitação do LMS nesse sentido, este deve dar resposta a dois níveis. Num primeiro, após validar o perfil do utilizador, deverá gerar de forma

automática, instâncias das estruturas de dados necessárias ao armazenamento da informação relativa, quer à “montagem” do curso (repositório de actividades), quer à sua utilização posterior (repositório de eventos).

Num segundo nível, deve colocar à disposição do “Professor” um conjunto de funcionalidades que lhe permitam definir e agendar as actividades concretas do curso, a associação dos conteúdos necessários, e a eventual activação de *chats* e/ou *fora* de discussão.

A figura 23 apresenta o diagrama de casos de uso correspondente a este âmbito de interacção com o sistema (LMS).

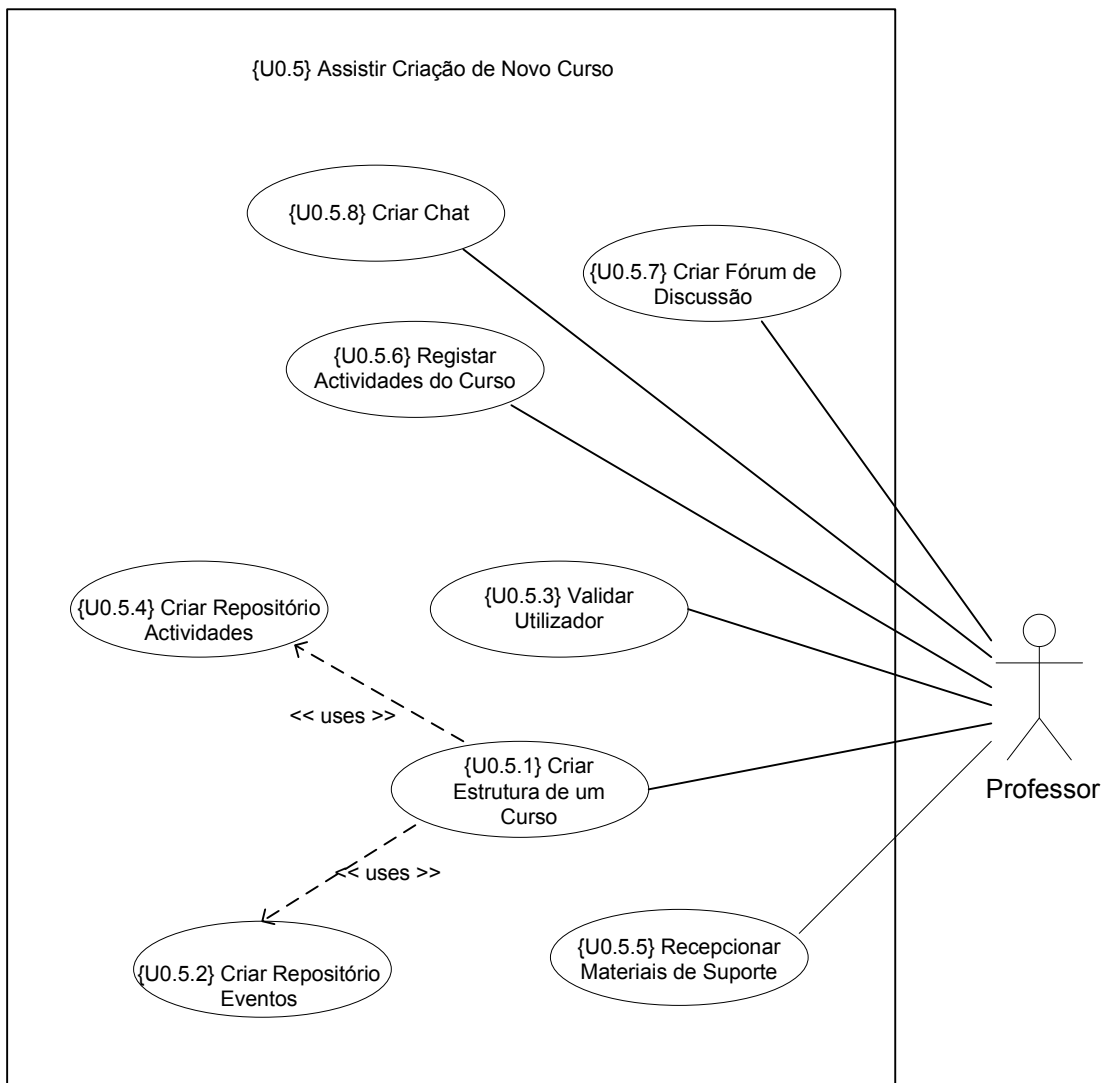


Figura 23 - Diagrama de Caso de Uso {U0.5} Assistir Criação de Novo Curso.

A figura 24 mostra o diagrama de sequência correspondente ao diagrama de casos de uso da figura 23.

Mais uma vez se fez uso do bloco *loop*, para enquadrar a tarefa repetitiva de definição das diversas actividades do curso, e de lhes associar, eventualmente, materiais de suporte.

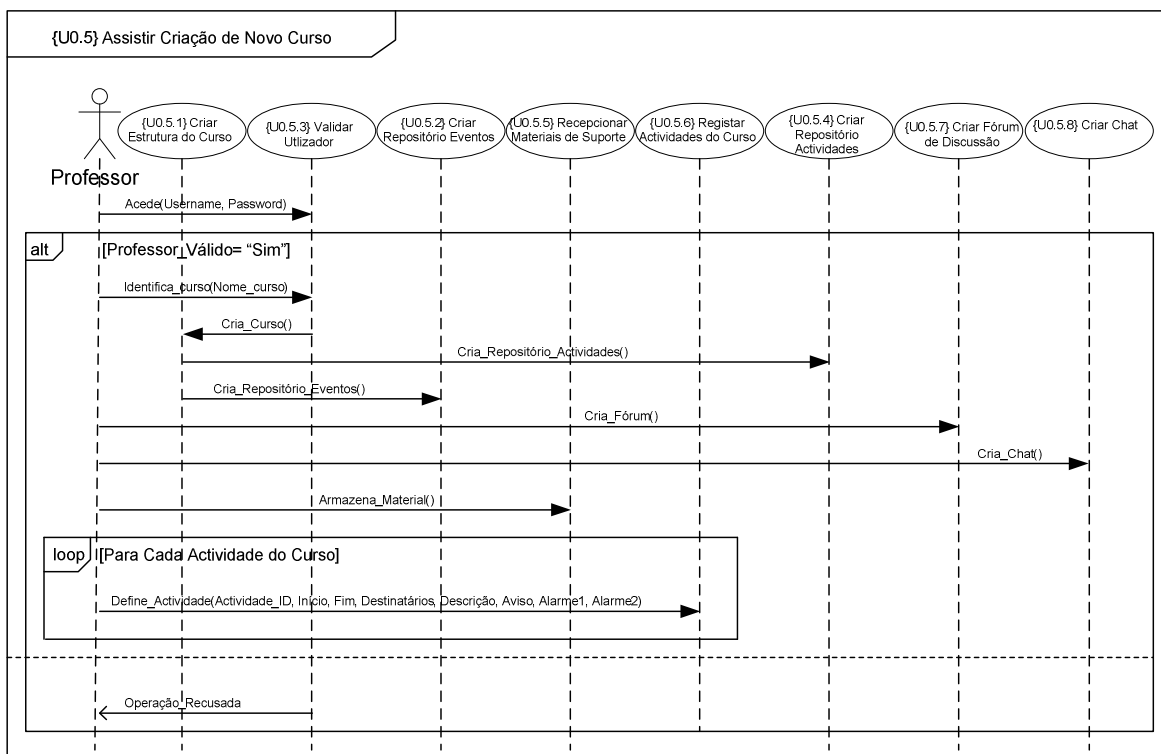


Figura 24 - Diagrama de Sequência do Diagrama de Casos de Uso {U0.5} Assistir Criação de Novo Curso.

### {U0.8} Suportar Execução de Actividades

Descreve-se seguidamente a interacção que todos os actores do tipo “Actor” têm com o sistema (LMS), no âmbito da execução de actividades.

Após serem validados enquanto utilizadores com as permissões necessárias, apenas terão que escolher um curso em que se encontrem inscritos e navegarem pelas actividades que o sistema (LMS) lhes for disponibilizando.

A figura 25 mostra o diagrama de casos de uso correspondente a este tipo de interacção.

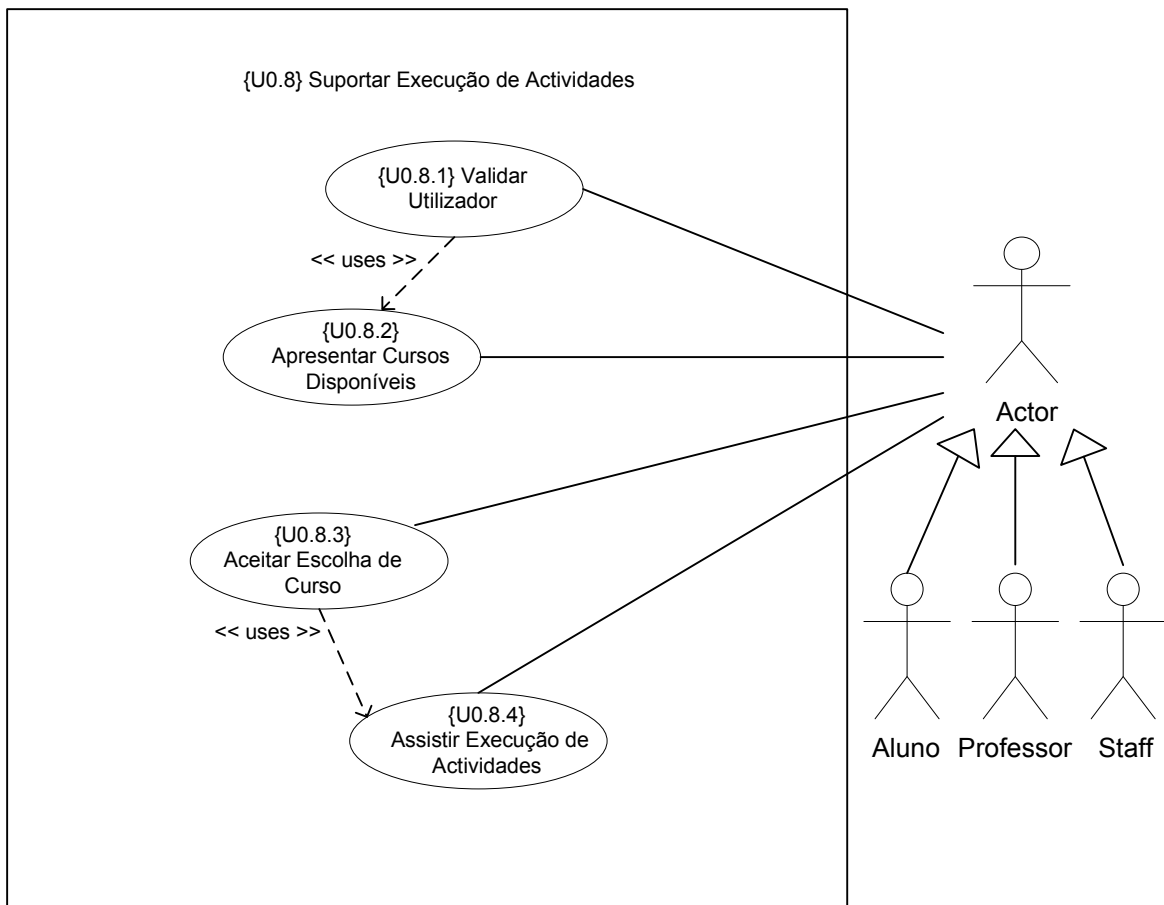


Figura 25 - Diagrama de Caso de Uso {U0.8} Suportar Execução de Actividades.

Na figura 26 é apresentado o diagrama de sequência correspondente ao diagrama de casos de uso “{U0.8} Suportar Execução de Actividades”, sendo talvez de entre todos os que se apresentaram até agora, aquele cuja sequência é menos fácil de acompanhar, sem algumas explicações suplementares.

Em cada sessão, o “Actor” começa por se submeter a uma validação e, caso reúna as condições para poder frequentar algum dos cursos disponíveis, ser-lhe-á dada a possibilidade de escolher o curso que quer frequentar. O LMS activará o curso escolhido e dois cenários de utilização estarão disponíveis: Gestão total das actividades por parte do LMS ou selecção das mesmas por parte do utilizador, com base numa lista actualizada dinamicamente pelo sistema, função do percurso já realizado pelo utilizador. Em qualquer dos casos, o utilizador poder-se-á ver envolvido num processo iterativo de execução das actividades que ele vai escolhendo, ou que lhe forem sendo apresentadas pelo LMS. Este

processo terminará quando o LMS não puder apresentar novas actividades, não houver mais actividades por executar, quando for atingido o limite temporal de execução do curso ou quando o utilizador abandonar a sessão.

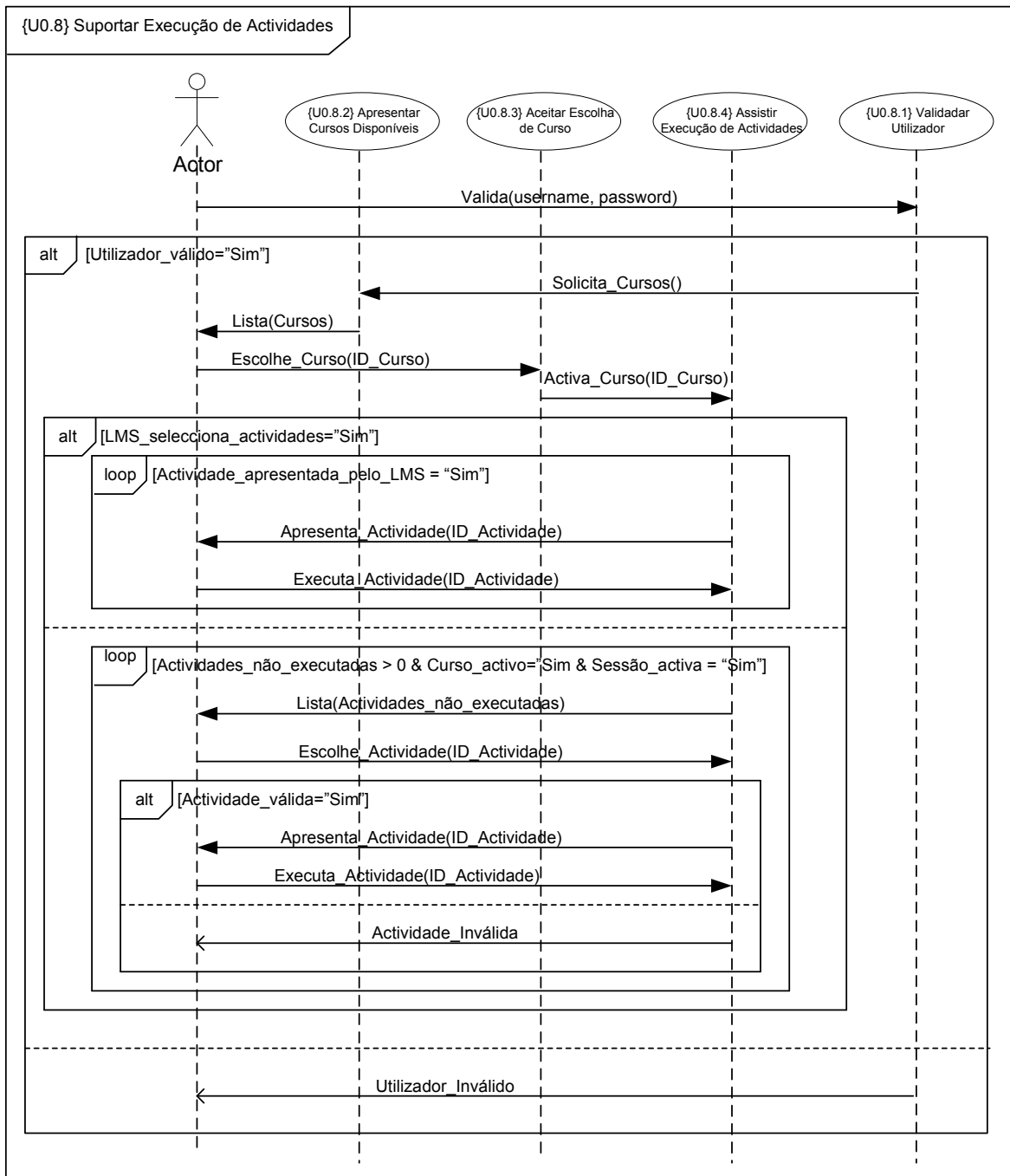


Figura 26 - Diagrama de Sequência do Diagrama de Casos de Uso {U0.8} Suportar Execução de Actividades.



### 5.1.3 Outras representações complementares

Tal como afirmado no início da secção 4.1, a actividade de criação do modelo de um sistema só terá a ganhar com a utilização de variadas e complementares técnicas de representação do conhecimento. Da conjugação de todas essas representações resultará sem dúvida um melhor entendimento do funcionamento dos sistemas representados, e a validação da captura do conhecimento realizada também sairá beneficiada já que a mesma será suportada num conjunto mais alargado de formas de representação de uma mesma realidade. Esse facto permite cruzar a informação recolhida através das diferentes técnicas de criação de modelos e, dessa forma, garantir uma representação final mais próxima da realidade.

#### 5.1.3.1 Estrutura hierárquica de navegação

A figura 27 mostra uma estrutura hierárquica cujo objectivo consiste na sistematização das diversas interacções que podem ser encontradas entre os diferentes actores e os LMS e que poderia perfeitamente representar uma estrutura de menus de um LMS.

Partindo do acesso que qualquer utilizador pode realizar, desde que conhecido o URL do sítio onde está instalado o LMS que suporta os cursos, é possível navegar pela referida estrutura, assumindo sucessivamente os diversos papéis previstos no contexto deste trabalho. Isto é, o utilizador pode aceder como visitante, não precisando para tal de qualquer validação e podendo apenas aceder a informação pública (por exemplo, consultar que cursos estão disponíveis na plataforma) ou solicitando o registo no sistema.

Por outro lado, tratando-se de um utilizador já registado, após solicitar a sua validação, pode apresentar-se ao sistema como aluno, como professor ou como elemento das equipas de suporte.

Como Aluno pode solicitar a inscrição num curso, executar actividades de um curso em que se encontre inscrito, comunicar síncrona ou assincronamente com os outros elementos do mesmo curso, aceder a conteúdos, fazer *downloads* ou *uploads* de ficheiros, etc.

Como Professor pode criar e redefinir cursos, executar actividades no âmbito de um determinado curso, monitorizar as participações dos outros actores, nomeadamente dos alunos, poderá criar fóruns de discussão, comunicar síncrona ou assincronamente com os outros actores, etc.

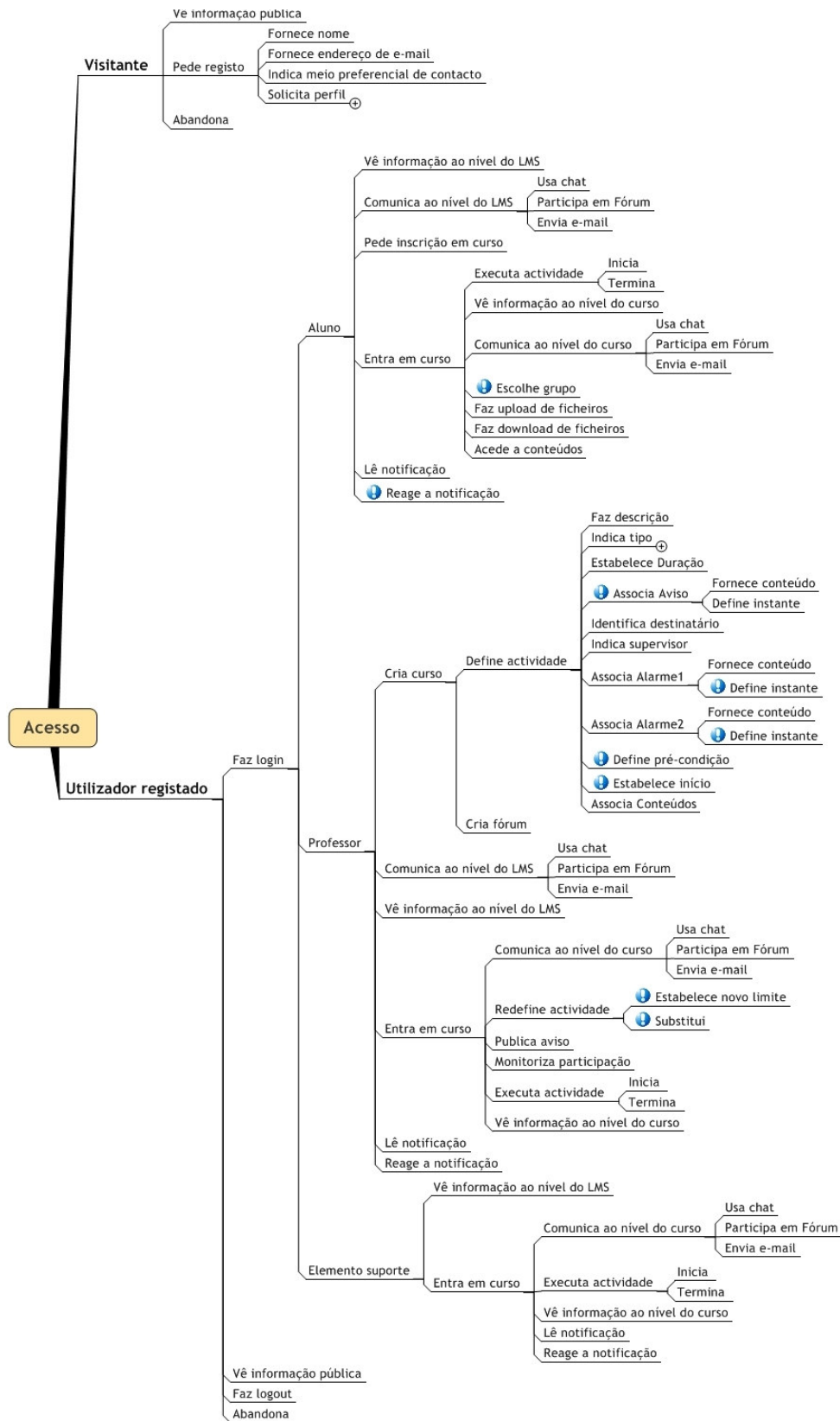


Figura 27 – Estrutura hierárquica de navegação num LMS.

Como elemento das equipas de suporte poderá executar actividades no âmbito de um determinado curso, comunicar síncrona ou assincronamente com os outros actores, etc.

### 5.1.3.2 Modelo conceptual em camadas

A figura 28 é uma outra representação, bastante sintética, das relações que podem estabelecer-se entre os diversos utilizadores e os LMS, sendo estes representados como uma estrutura de camadas a cujas funcionalidades podem aceder apenas os utilizadores com o perfil adequado.

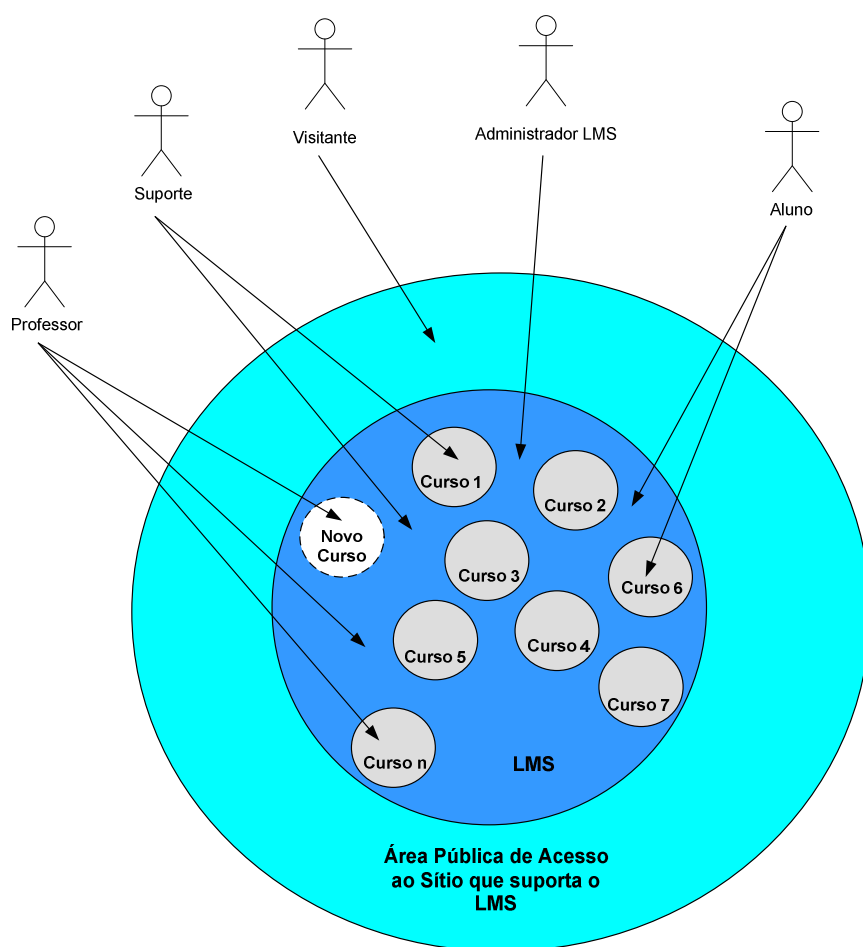


Figura 28 – Representação conceptual em camadas, das interacções com um LMS.

Observando o diagrama da figura 28, fica-se com a impressão de que mais nenhum utilizador para além do Visitante pode aceder à camada “Área Pública de Acesso ao Sítio que suporta o LMS”. No entanto, a leitura correcta que se deve fazer deste diagrama,

usando o exemplo acabado de referir, é que qualquer outro utilizador (Professor, Aluno, Suporte ou Administrador), quando acede à camada em questão, ainda não teve que efectuar qualquer validação no sistema e, portanto, é visto como um Visitante.

O diagrama da figura 28 dá uma ideia resumida do tipo de permissões que cada perfil de utilizador dá a quem o possui.

## 5.2 O problema ao nível da gestão das actividades

### 5.2.1 Introdução

A secção 1.2.1 deste texto introduziu o conceito básico em que assentou o desenvolvimento da maior parte do trabalho que se tem vindo a descrever - a actividade.

Foram identificados os metadados necessários à completa descrição de uma qualquer actividade e foi possível confirmar a estrutura da unidade elementar de gestão de actividades de ensino/aprendizagem em tempo real, apresentada na figura 2, como sendo a que melhor poderia servir os interesses subjacentes ao trabalho a levar por diante.

De acordo com tal estrutura, um curso é, então, composto por actividades com um início predefinido ou não, a executar por um ou mais actores (alunos, professores e elementos da equipa de suporte), que devem ser executadas dentro de janelas temporais com uma duração prevista.

Antes de iniciada qualquer actividade, a camada de gestão deve emitir um aviso relativo à proximidade de início da mesma e, antes de esgotado o tempo previsto para a sua execução, terá que verificar se a actividade foi executada ou está em curso. Se nenhum dos casos se verificar, deverá ser emitido um primeiro alarme. Desta forma, pode evitar-se ter que voltar a programar as actividades do curso. Em todo o caso, após esgotado o tempo previsto para a execução da actividade, o camada de gestão, terá que voltar a verificar se a actividade foi executada. Em caso negativo, os intervenientes relevantes devem ser notificados através do envio de um segundo alarme.

Os procedimentos a adoptar perante uma situação de alarme são específicos de cada caso concreto. Da mesma forma, os espaços temporais que separam um aviso do início de uma actividade, a emissão de um primeiro alarme do limite de execução da actividade e o

fim de uma actividade, da emissão do segundo alarme, são também variáveis e a definir em cada situação.

Nunca se considerou que fosse uma competência da camada de gestão, qualquer tipo de controlo da execução dos procedimentos correctivos. Isto é, não cabe a esta camada garantir que as acções a executar, como resultado do não cumprimento de uma actividade, são efectivamente executadas e se o são da forma prevista. À camada de gestão compete apenas detectar a necessidade de notificar e iniciar o processo de notificação dos actores relevantes, no contexto da actividade não executada.

A identificação de uma estrutura elementar como a que se apresentou na figura 2 foi um passo importante na realização do projecto. Contudo, era necessário prosseguir no sentido do aprofundamento da caracterização do problema, como forma de chegar a uma proposta de metadados de gestão satisfatória.

Relembremos que na origem da realização deste trabalho estavam preocupações de eventual integração com o trabalho já disponível no âmbito dos projectos IMS e/ou SCORM.

Da reflexão que foi sendo levada a cabo sobre o tema, resultou a constatação de que o modelo apresentado na figura 2 seria potencialmente bom, mas que existiam situações que obrigavam a manipulações particulares do mesmo. Por exemplo, nem sempre é possível identificar o valor do atributo “instante inicial” da actividade. Na realidade, verificou-se que existem dois tipos de actividades básicas. Actividades com início predefinido e actividades de execução aleatória.

Novos aspectos foram sendo identificados e em consequência disso, houve necessidade de introduzir adaptações ao modelo original. Por exemplo, num cenário de um curso com execução aleatória de actividades, deixaria de poder considerar-se a emissão de avisos prévios, já que é impossível saber que actividade será escolhida pelo utilizador em cada momento. Esta situação deriva de um modelo de selecção das actividades que coloca a ênfase no actor e não no LMS. Ou seja, não é qualquer mecanismo fazendo parte do LMS que se encarrega de seleccionar a próxima actividade a apresentar ao actor.

Aquele que parecia ser inicialmente o aspecto fulcral para a resolução do problema, as notificações e os sistemas que as possibilitariam, rapidamente perdeu importância relativamente face a outros aspectos do processo. Essa perda de importância relativa deveu-se ao facto de se ter concluído que, fosse qual fosse a situação que motivasse a

necessidade de uma notificação (aviso, informação ou alarmes), apenas havia que invocar os mecanismos de *messaging* do LMS, passando-lhes implícita ou explicitamente, o conteúdo da mensagem a enviar, bem como a identificação dos destinatários e o meio de comunicação preferencial a utilizar em cada caso (endereços de correio electrónico, números de telefone, endereços postais, etc).

Tão importante como identificar as actividades específicas, foi distinguir os tipos de actividades que podem ser encontradas, quer no que toca ao instante temporal em que ocorrem, quer no que diz respeito ao processo que leva à sua selecção, ou ainda no que se refere ao nível de dependência de uma actividade em relação a outras, ou à decomposição de uma actividade em subactividades.

Para além destas situações particulares, houve que reflectir sobre as consequências que podem resultar de se inserir, remover ou reposicionar uma actividade num curso já em execução.

Uma reflexão aprofundada a propósito dos diversos aspectos que se acabou de referir, bem como da adopção de TIC de suporte a processos de ensino/aprendizagem a distância, e do envolvimento dos diversos actores que nesse âmbito é possível encontrar, levou à identificação do seguinte conjunto de tipos de actividades:

- a) Actividades sequenciais;
- b) Actividades paralelas;
- c) Actividades de execução aleatória;
- d) Actividades compostas de subactividades;
- e) Actividades condicionadas;
- f) Actividade de constituição de grupos.

Ao mesmo tempo, identificaram-se também as seguintes situações, merecedoras de uma análise especial:

- a) Alteração da ordem das actividades;
- b) Execução de actividades por grupos;
- c) Execução de actividades ao longo de diversas sessões;
- d) Execução aleatória de actividades;
- e) Execução condicionada de actividades;
- f) Execução de actividades eventuais.

Ver-se-á de seguida, com algum detalhe, cada uma das situações identificadas.

## 5.2.2 Tipos de actividades

Como tem sido referido, o presente trabalho deu continuidade ao projecto ODL Toolbox. Da realização desse projecto resultou a identificação de cerca de oitenta actividades, distribuídas pelas quatro fases de um curso – “Concepção”, “Planeamento”, “Execução” “Avaliação” (Ramos et al, 2003).

Enquanto que no projecto ODL Toolbox houve a preocupação de se identificarem, o mais exaustivamente possível, as actividades passíveis de ser encontradas durante a execução de um curso, no presente trabalho deu-se um salto qualitativo de considerável significado. Em vez de se tentar identificar as actividades, procurou-se isolar e caracterizar, mais abstractamente, os tipos de actividades que poderão ser encontrados em qualquer curso a distância, suportado por um LMS. Desta forma, o esforço foi orientado para a identificação dos atributos informacionais que podem caracterizar uma qualquer actividade, independentemente da fase do curso em que ocorra.

Não se pretende com o que acabou de ser dito, levar o leitor a pensar que a abordagem seguida no projecto ODL Toolbox terá constituído uma má opção. Antes pelo contrário. Relativamente às actividades em concreto, é importante a sua identificação, dado que só dessa forma será possível complementá-las com os avisos, informações e alarmes adequados.

Far-se-á de seguida a apresentação e descrição dos tipos de actividades identificados, sendo certo que não se exclui a possibilidade de esta lista poder ser aumentada, quer como fruto de uma maior reflexão, quer pela contribuição de terceiros, capazes de identificar novas situações que não foi possível descobrir durante a realização do presente trabalho.

### 5.2.2.1 Actividades sequenciais

Este tipo de actividades é um dos que mais utilização tem na construção de cursos suportados por LMS. Caracterizam-se pelo facto de cada actividade apenas ser apresentada para execução após estar terminada a actividade anterior. Isto é, qualquer que seja a actividade  $i$ , com  $i > 1$ , o instante de início da actividade  $i$  terá que ser posterior ao instante de finalização da actividade  $i-1$ .

Nestes casos, o modelo de unidade atómica de gestão de actividades aplica-se perfeitamente. Um curso será neste caso, uma sequência de unidades elementares ou atómicas como a que se representou na figura 2.

A gestão do lançamento de cada uma das actividades será efectuada pelas outras camadas dos LMS. À luz do projecto SCORM, tal missão poderia caber à componente *Sequencing and Navigation*, sendo que esta, por sua vez se baseia no modelo *Simple Sequencing* do IMS (IMS, 2003c). Poder-se-á apresentar a título de exemplo o caso representado pela execução da actividade “inscrição em curso” e pela execução da primeira das actividades desse curso. Nenhum aluno poderá executar qualquer actividade de um curso, antes que esteja inscrito no mesmo. Notar que a inscrição de um aluno num curso pode ser considerada como a primeira actividade a executar no âmbito desse curso!

#### 5.2.2.2 Actividades paralelas

O tratamento das actividades paralelas também não apresenta qualquer tipo de problemas no âmbito da aplicação do modelo proposto.

Actividades que devem ser executadas paralelamente caracterizam-se por possuírem um período comum quanto à janela temporal de execução. Esse período pode ser parcial, total ou até, o tempo previsto para execução de uma actividade, ser de tal forma longo que possibilite o lançamento de várias actividades diferentes nesse mesmo período de tempo. A ocorrência de actividades de execução paralela não implica a necessidade de qualquer tratamento especial relativamente à utilização do modelo de actividade apresentado na figura 2.

Também neste caso continuamos a ter um início da actividade, uma janela temporal de execução, um instante final, e a possibilidade de associação de um aviso, de informação, de um primeiro alarme e de um segundo alarme, notificações a serem enviadas antes do instante inicial da actividade, antes de atingido o instante definido para o primeiro alarme, antes de atingido o limite temporal para execução da actividade e após o instante final da mesma.

Imagine-se uma situação em que o autor (professor) tenha definido uma actividade consistindo na escrita de um texto a submeter até ao fim do curso, relativo a um tema disponibilizado e estudado como uma actividade lançada na fase inicial do curso. Neste



caso, desde o aparecimento dessa actividade até à data limite para a submissão do texto, poderão ocorrer inúmeras actividades que, portanto, decorrerão em paralelo.

Uma outra situação que pode ser apresentada como exemplo deste tipo de actividades, muito vulgar em termos do funcionamento de cursos a distância, consiste no lançamento de uma actividade de estudo de uma determinada matéria que pode culminar com a apresentação de um trabalho sobre a mesma. Durante o tempo em que essa actividade decorre, podem ser lançadas sessões de comunicação síncrona entre os diversos alunos e o professor, eventualmente via *chat*, para discussão em tempo real de determinados temas chave fazendo parte da matéria em estudo. Tais sessões, sendo elas também actividades programadas, decorrem, pois, em paralelo com a actividade principal.

### 5.2.2.3 Actividades de execução aleatória

Este tipo de actividades caracteriza-se pela impossibilidade de estabelecer à partida um instante de início para a execução das mesmas.

Neste contexto, abdica-se das funcionalidades associadas ao subsistema de avisos da camada de gestão, já que não sendo conhecido o instante inicial das actividades, também não é possível definir as condições temporais para a emissão dos avisos correspondentes.

Contudo, não é difícil identificar-se um cenário no qual o curso ou parte dele consiste num conjunto de actividades cuja ordem de execução não é relevante. A única condição que tem que ser verificada é que todas as actividades sejam executadas dentro dos limites temporais estabelecidos para o curso, isto é, até que seja atingida a data de conclusão do mesmo.

Esta condição, pode ser verificada de dois modos diferentes:

- a) Cada actividade terá que ser executada durante o tempo previsto para a sua execução. Ou seja, após ter escolhido e iniciado uma actividade, o actor terá que a concluir nesse espaço de tempo. Se tal não acontecer, a camada de gestão emitirá um segundo alarme.
- b) A camada de gestão só verifica a conclusão de todas as actividades quando for atingido o instante definido para o fim do curso.

No segundo caso, o mecanismo de alarmes também não tem qualquer utilidade já que quando for concluída a necessidade de uma eventual emissão de uma notificação desse tipo, já o curso terá terminado. Esta situação corresponderia a contrariar o princípio base de que se partiu para a realização deste trabalho. Relembre-se que a ideia fulcral consistiu em obter um sistema de monitorização capaz de reagir automaticamente e em tempo útil a desvios, face aos programas ou planos dos cursos.

Face ao que se acabou de ver, embora não sendo impossível a utilização do modelo proposto, essa utilização implica o desaproveitamento de algumas das características do mesmo.

Poder-se-á pensar num contexto específico em que o sucesso no curso é atingido se for possível concluir que cada uma das actividades do mesmo, dizendo respeito ao estudo e avaliação de temas distintos e independentes, foi executada com sucesso antes de atingido o limite temporal para a conclusão do curso.

Trata-se de uma situação em que a ordem pela qual as actividades são executadas, não é relevante para o processo de ensino/aprendizagem. Apenas importa que todas elas estejam terminadas quando o curso atingir o seu limite temporal.

#### 5.2.2.4 Actividades compostas de subactividades

O tratamento de actividades compostas por subactividades pode também ser encontrado no âmbito do IMS LD (IMS, 2003a). Aí, este tipo de actividades é materializado em estruturas designadas por *activity structures* que invocam outras estruturas de actividades ou mesmo unidades de aprendizagem (*Units of Learning*) externas.

Ver como se comportaria o modelo proposto, em situações deste tipo, foi a preocupação que se seguiu. Conceptualmente não existe qualquer diferença, já que apenas se passa a ter a invocação de um outro conjunto de actividades (subactividades), cada uma com uma janela temporal de execução, um aviso e alarmes associados.

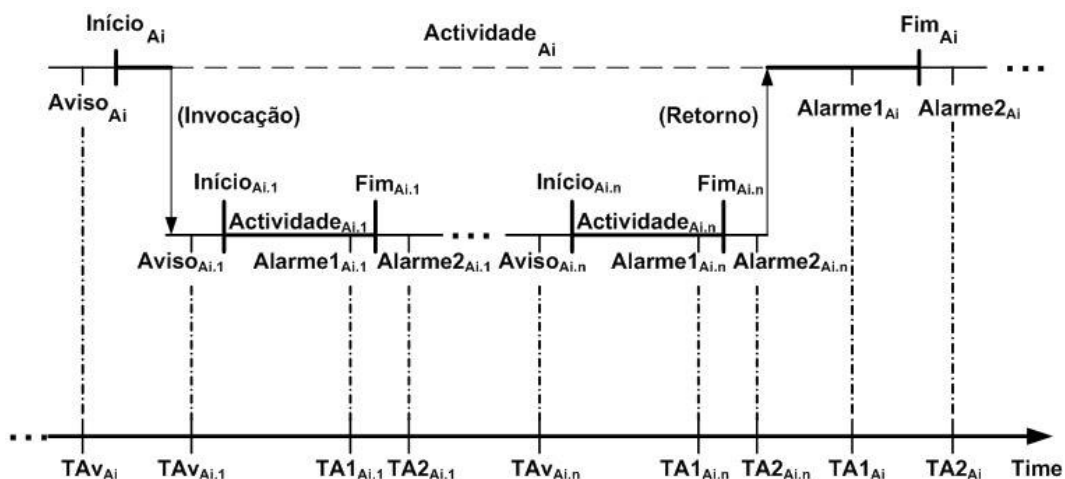


Figura 29 – Invocação de uma estrutura com duas subactividades a partir de uma actividade de nível superior (actividade pai).

Apenas haverá a necessidade de garantir que o instante definido para o aviso da primeira das subactividades seja posterior ao início da actividade invocadora e que o instante definido para o segundo alarme da última subactividade seja anterior ao instante definido para o primeiro alarme associado à actividade a partir da qual se esteja a fazer a invocação do conjunto das subactividades. Ou seja:

$$T_{AvAi.1} > T_{Início Ai} \quad e \quad T_{A2Ai.n} < T_{A1Ai}$$

O esquema da figura 29 ilustra esta situação.

Torna-se claro a partir da observação da figura 29 que em determinado instante a actividade  $A_i$  é suspensa e é feita a invocação das subactividades  $A_{i.1}$  a  $A_{i.n}$ . Após a execução da última das subactividades ( $A_{i.n}$ ), a actividade  $A_i$  é retomada, prosseguindo a execução das actividades previstas no nível superior.

Pareceu ser interessante ainda a este nível, retratar em termos dos elementos de metadados, o posicionamento relativo de uma subactividade face às outras. A solução passou pela inclusão de um atributo “actividade\_pai”, associado à entidade informacional “Actividade”, destinada à indicação da actividade da qual deriva ou faz parte. Tratando-se de uma actividade simples, esse atributo será preenchido com o identificador da própria actividade. Este mecanismo pode ser utilizado em decomposições multi-nível.

De referir ainda o facto de não ser necessária qualquer tipo de preocupação especial com a emissão dos alarmes por incumprimento ou atraso em determinada subactividade, já

que a justificar-se, o mesmo poderá ser emitido, quer no instante definido para fim da subactividade não cumprida, quer no instante definido para os alarmes associados à actividade pai.

De realçar o facto de uma estrutura de actividades, tal como prevista no *IMS Learning Design*, poder modelar uma sequência ou uma selecção de actividades. A possibilidade de selecção coincide com o cenário apresentado acima, relativo à execução aleatória de actividades.

O exemplo apresentado para a execução de actividades aleatórias pode ser recuperado para este caso.

Tome-se cada uma das actividades referidas, dizendo respeito a um tema de estudo específico, e composta por uma subactividade de estudo de determinado conteúdo, e uma outra de avaliação dos conhecimentos adquiridos sobre esse tema. A actividade pai poderia ser caracterizada pelos seguintes dados:

**Identificação: Actividade<sub>i</sub>**

Designação : “Aceder ao conteúdo X e após estudá-lo, executar a actividade de avaliação respectiva”.

Início : Dia d, hora 0

Duração : 5 dias

Fim : Dia d + 5, hora 24

Supondo que as subactividades eram de execução sequencial, poderiam apresentar as seguintes características:

**Identificação: Actividade<sub>i,1</sub>**

Designação : “Aceder ao conteúdo X e estudá-lo”.

Início : Dia d, hora 0

Duração : 4 dias

Fim : Dia d + 4, hora 24

**Identificação: Actividade<sub>i,2</sub>**

Designação : “Executar o teste de avaliação relativo ao conteúdo X”.

Início : Dia d + 5, hora 0

Duração : 1 dia

Fim : Dia d + 5, hora 24

### 5.2.2.5 Actividades condicionadas

À medida que o trabalho foi avançando, novas situações foram emergindo e, embora não sendo novas em termos do senso comum, julgou-se inicialmente que estariam fora do contexto deste trabalho. Por exemplo, será que devia ser contemplado o caso da execução condicionada de actividades? Se sim, esse facto implicaria a necessidade de incluir novos elementos de metadados ou atributos, adequados à representação e tratamento desses casos?

Este aspecto não diz directamente respeito à camada de gestão, já que é às outras camadas do LMS que cabe a tarefa de identificar se estão reunidas as condições para o lançamento de uma nova actividade. Não seria de estranhar que essa função estivesse associada à componente que trata do sequenciamento e navegação<sup>48</sup>. No entanto, uma situação deste tipo acaba por ter implicações ao nível da camada de gestão, já que será necessário que esta decida sobre se deve ou não ser emitido o aviso relativamente à proximidade do início de uma actividade com essas características. Com efeito, se não estiverem reunidas as condições para que o destinatário da actividade a possa executar, não faz sentido emitir o aviso relativo à mesma.

A situação, porém, poderia ser aproveitada pela camada de gestão para avisar o actor da proximidade do início da actividade e para o alertar acerca das condições que estariam a ser violadas, impedindo-o de vir a executar a próxima actividade.

Não deve ser confundida esta função com aquela que caberá às outras camadas do LMS, consistindo em determinar se existem ou não condições para lançar uma dada actividade com destino a um determinado actor.

Na verdade, trata-se de um só problema mas visto e tratado sob perspectivas diferentes. Quer a camada de gestão, quer as outras camadas do LMS precisam de verificar se estão reunidas as condições para que a actividade seja lançada. O que difere de uma situação para a outra é o instante em que essa verificação é realizada.

Dado que o “Aviso” é enviado antes do lançamento da actividade, a camada de gestão precisa de verificar essa situação em primeiro lugar.

Por outro lado, ao incluir-se elementos e atributos de metadados destinados a retratar actividades cuja execução está dependente da verificação de determinadas condições, foi

---

<sup>48</sup> Em termos do projecto SCORM, o tema pode ser aprofundado no livro *Sequencing and Navigation Version 1.3*, disponível em <http://www.adlnet.gov/scorm/20043ED/Documentation.cfm>.

garantida a possibilidade de envio de alarmes mais ricos, uma vez que passou a ser possível identificar quais as condições que não se verificaram e que impediram a apresentação da actividade ao actor ou actores, ao mesmo tempo que se torna possível emitir avisos igualmente enriquecidos, como se descreveu anteriormente.

Apesar das considerações efectuadas, não restam quaisquer dúvidas quanto ao facto de ter de ser a componente de sequenciamento e navegação do LMS a decidir se a actividade deve ou não ser apresentada ao actor ou ao grupo de actores e de dever também ficar a cargo das outras camadas do LMS o registo da informação referente à finalização da actividade (implicando a actualização do repositório “Eventos”). À camada de gestão, continua a estar associada apenas a função de verificar no instante limite da actividade, por consulta ao repositório “Eventos”, se esta foi ou não executada.

Ver-se-á mais adiante, na secção “5.3 Funcionalidades da componente software”, mais aprofundadamente, como é que esse facto interfere com a decisão a tomar pela camada de gestão e de que forma foi resolvido o problema.

Facilmente se pode apresentar um exemplo deste tipo de actividades. Pense-se no caso em que a um aluno apenas será apresentada uma actividade se o mesmo tiver tido na actividade anterior, consistindo numa actividade de avaliação (realização de um teste, por exemplo), uma classificação igual ou superior a determinado valor predefinido.

#### 5.2.2.6 Actividade de constituição de grupos

Parece estranho que se tenha identificado esta actividade como justificadora de um tratamento particular. Contudo, ela apresenta aspectos que a tornam singular.

A designação da actividade pode ser enganadora. Não se trata de uma actividade de grupo mas sim uma actividade a ser executada por cada um dos alunos inscritos no curso.

Simultaneamente, estar-se-á em presença de uma actividade condicionada, já que é de prever a existência de regras definidas para a formação dos grupos. Por exemplo, o número máximo e mínimo de elementos de um grupo, ou o número máximo de grupos.

A constituição de grupos é uma actividade que normalmente diz respeito apenas aos alunos e, no âmbito de um curso, pode ser executada de duas formas. Num primeiro cenário o professor decide que elementos vão formar cada grupo e, neste caso, não existe qualquer dificuldade. Numa segunda situação, essa é uma actividade a ser executada

autonomamente pelos próprios alunos, sendo este o caso que mais interesse tem num contexto como aquele que presidiu à realização do presente trabalho.

Tratando-se de uma actividade observável e, portanto, passível de ser gerida, importava verificar se estavam reunidas as condições para que pudesse ser monitorizada pela camada de gestão em desenvolvimento e se seria possível enquadrá-la no modelo de gestão de actividades proposto.

Com efeito, pode-se estabelecer um instante inicial e uma janela temporal de execução para a actividade, sendo assim possível também, determinar o limite para a sua execução. Conhecendo-se o instante inicial, é possível definir com que antecedência deve ser emitido o aviso correspondente à sua execução e, determinado o limite para a sua execução, também se torna possível identificar os instantes relativos ao primeiro e ao segundo alarmes.

Não se adivinha, portanto, qualquer dificuldade na implementação das funcionalidades necessárias à gestão da execução de uma actividade deste tipo.

### 5.2.3 Operações sobre actividades

#### 5.2.3.1 Alteração da ordem das actividades

Durante o funcionamento de um curso é possível que, por diversos motivos, haja necessidade de introduzir alterações ao plano inicial do mesmo, isto é, promover inserções, remoções ou mudanças de posição de actividades dentro da estrutura global do curso.

Alguns dos trabalhos de normalização em curso (SCORM, IMS, etc) contemplam configurações dinâmicas da estrutura dos cursos, por exemplo, de acordo com o desempenho acumulado pelo aluno nas actividades anteriores.

A auto-configuração de cursos, função do desempenho dos alunos, é um tema actual fora do âmbito dos projectos de normalização que se tem vindo a referir (Jovanovic et al, 2006).

Na verdade, essas alterações não são mais do que adaptações à combinação de actividades que constituem o plano do curso, originando por cada alteração, uma nova configuração desse plano, já que não é de esperar que todos os actores falhem a execução

de uma dada actividade e, por esse motivo, não se justificaria reformular o curso para os intervenientes cumpridores.

Pensando na resposta que o modelo apresentado neste trabalho daria em situações como essa, conclui-se que apenas existiria um problema caso uma actividade sofresse qualquer tipo de alteração já depois do aviso correspondente ter sido emitido. A inclusão de uma restrição a esta operação em tais circunstâncias resolveria o problema. Tal restrição não é difícil de implementar, a partir do momento em que está disponível um atributo associado ao elemento de metadados “Actividade”, destinado a registar o facto de o aviso associado, já ter sido enviado ou não.

Em casos extremos e mais recentemente equacionados (dada a evolução da tecnologia), pode mesmo pensar-se na auto-configuração de cursos em função da localização do aluno. Nesses casos, utilizando capacidades de computação móvel, e de Sistemas de Informação Geográficos, os LMS poderão procurar os serviços de aprendizagem, os locais, os elementos de suporte e a informação mais adequados à localização e estado de evolução de cada aluno (Yang, 2006).

Imagine-se, por exemplo, um curso com uma componente de trabalho de campo destinada à análise geológica do local em que o aluno se encontrasse. Seria possível apresentar, em cada caso, conteúdos relevantes à realização do trabalho, em função da identificação da localização dos alunos.

### 5.2.3.2 Execução de actividades por grupos

Um dos cenários possíveis no âmbito da utilização de LMS para suporte a acções de ensino/aprendizagem, e que se assume como dos mais complexos, é o da execução de actividades, simples ou compostas, por grupos de actores que, tipicamente serão alunos.

As actividades de grupo mereceram uma atenção especial, já que podem estar directamente relacionadas com conceitos e práticas que no âmbito dos processos de ensino/aprendizagem à distância assumem um papel de destaque. É o caso do trabalho colaborativo e da aprendizagem colaborativa. Por outro lado, trata-se de uma situação não coberta pela especificação SCORM, o que confere ao trabalho realizado nesta vertente da arquitectura dos LMS, uma importância muito significativa, uma vez que se assume como um potencial melhoramento à especificação referida.



Na sua versão menos problemática, uma actividade de grupo será simples e todos os elementos do grupo serão confrontados com a mesma actividade. Trata-se de um cenário pouco previsível e de tratamento exactamente igual ao que já foi abordado para os outros casos. Isto é, cada elemento do grupo é avisado da proximidade da actividade e dispõe de um intervalo de tempo para a executar. Caso a não execute dentro do prazo previsto, receberá do sistema, bem como os outros destinatários definidos, uma notificação do tipo primeiro alarme ou segundo alarme, dependendo do momento da constatação de tal incumprimento.

Numa outra situação, mais complexa e também mais próxima dos casos reais, a actividade é composta de subactividades e cada elemento do grupo deve executar uma das subactividades, sem contudo se determinar à partida que elemento do grupo deve executar que subactividade.

Este cenário não é contudo incompatível com o modelo proposto. Note-se que se está perante um caso particular de actividades de execução aleatória. Agora, o carácter aleatório de execução das actividades não se refere, em princípio, ao instante de início das mesmas, mas sim aos actores que as vão executar, embora possamos considerar as duas situações a ocorrerem em simultâneo. Teremos neste caso, execução aleatória das subactividades e selecção aleatória da subactividade que cada elemento do grupo vai executar.

Os únicos atributos que estariam por preencher seriam, assim, a identificação do actor destinatário da actividade, bem como o instante inicial da mesma e, portanto, também o instante de emissão do aviso correspondente. Não sendo possível conhecer qual o actor que vai executar a subactividade, também se tornaria impossível o envio do aviso correspondente.

O preenchimento do atributo “executante” da actividade seria efectuado pelas outras camadas do LMS logo que um actor escolhesse uma subactividade.

Um mecanismo semelhante àquele que se indicou para a actividade de constituição de grupos permitiria tratar este tipo de situação de uma forma eficaz.

Na prática, o que existe é um conjunto de ofertas (subactividades) destinadas a ser escolhidas por um conjunto de actores (alunos de um grupo). Cada aluno de um grupo seria confrontado com a lista das subactividades ainda não atribuídas e escolheria uma delas. O sistema (LMS) se encarregaria de retirar da lista a subactividade escolhida, para que um outro elemento do grupo deixasse de a visualizar.

Nestes casos o único aviso que seria emitido seria o da actividade pai, alertando para a necessidade de os elementos dos grupos escolherem as subactividades. Após estas serem escolhidas, para cada uma delas, tudo se passaria como nos casos já descritos.

### 5.2.3.3 Execução de actividades ao longo de diversas sessões

Trata-se de um aspecto que do ponto de vista da componente de sequenciamento e navegação dos LMS tem uma grande importância, mas que ao nível da camada de gestão perde toda e qualquer relevância. Com efeito, o facto de uma actividade não ser executada apenas numa sessão não interfere em nada com as preocupações de gestão que estiveram na base deste trabalho. À camada de gestão apenas importa verificar o estado de execução da actividade quando o limite temporal predefinido é atingido. Tal verificação consistirá em consultar o repositório de “eventos” e verificar se existe um evento correspondente à actividade. Se existir, certamente que a actividade foi executada dentro da janela temporal prevista, caso contrário estarão reunidas as condições para a emissão de um segundo alarme.

Outra forma de tratar o problema consistiria em verificar o instante em que a actividade tivesse terminado. Nesse caso o LMS registaria sempre a execução da actividade (no repositório “eventos”), ainda que a mesma tivesse sido terminada para além do limite previsto para a sua execução. A camada de gestão detectaria a situação de igual modo e, desta forma, o sistema poderia conter mais informação com interesse em termos de histórico.

Mais uma vez, será às outras camadas do LMS que competirá promover o registo do evento quando a actividade tiver sido terminada.

### 5.2.3.4 Execução aleatória de actividades

Um curso constituído, total ou parcialmente, por actividades de execução aleatória, é algo que pode ocorrer. Que tratamento especial deve merecer tal cenário?

Como se viu já, actividades de execução aleatória caracterizam-se pela impossibilidade de ser definido um instante inicial para execução das mesmas e, por esse facto, não ser possível também a definição de notificações do tipo aviso a serem enviadas antes do início da sua execução.

Contudo, também neste caso, parece não existir qualquer impedimento à utilização do modelo que se propõe no presente trabalho. Com efeito continua a existir um conjunto de actividades que se iniciam em determinado instante e, com base nesse instante e na duração máxima definida para a actividade, o LMS pode determinar o instante limite para execução da mesma, actualizando o atributo correspondente. Em consequência disso, a camada de gestão pode realizar a sua tarefa de controlo, emitindo os alarmes eventualmente necessários.

### 5.2.3.5 Execução condicionada de actividades

Este tema foi já abordado anteriormente, do ponto de vista da caracterização desse tipo de actividades. Dever-se-á no entanto pensar também no que deverá ser o funcionamento desejável dos LMS, no tratamento de tais actividades.

Embora tratando-se de competências das camadas de sequenciamento e navegação dos LMS, não deixa de ser interessante e até importante, estabelecer alguns referenciais de procedimentos que devem ser verificados. Mais adiante se verá de que forma este facto se revelará importante no que à camada de gestão diz respeito.

O aspecto que parece merecer maior relevância prende-se com o facto de, apesar de ser possível identificar diferentes situações de ocorrência de actividades condicionadas, existir uma propriedade comum a todas elas – apenas deverão ser apresentadas para execução caso haja sucesso no teste das condições que lhes estão associadas. Ter-se-á pois, em qualquer caso, uma situação representável por:

Apresentar actividade se condição<sub>1</sub> e condição<sub>2</sub> e ... e condição<sub>n</sub>

Em que cada condição<sub>i</sub> implica a existência de um triplo composto de um identificador único, um texto descrevendo a condição a testar e um atributo de estado, indicador de se a condição se verificou ou não. Obviamente, deverão ser as outras camadas do LMS a actualizar o atributo de estado associado a cada condição.

Ao incluir no modelo global estes aspectos, tornou-se possível a emissão de notificações mais ricas. De facto, passa a ser possível a emissão de notificações aquando da não execução de uma actividade condicionada, com a inclusão dos motivos para a não apresentação da actividade ao actor ou actores.

Não deve ser esquecido que uma actividade deste tipo pode reunir as condições para ser apresentada pelo LMS e acabar por não ser executada dentro da janela temporal predefinida. Caberá à camada de gestão distinguir as duas situações e emitir a notificação adequada a cada caso.

### 5.2.3.6 Execução de actividades de recuperação

Até agora apenas se tem feito referência a actividades que devem ser executadas obrigatoriamente para que um curso possa ser considerado concluído. Existem contudo situações que aconselham a inclusão do que se poderia chamar actividades de recuperação. Estas não constituem de facto um tipo diferente de actividades, tornando-se relevantes apenas por fazerem parte de mecanismos de recurso destinados a corrigir desvios face ao plano previsto para o curso.

Na eventualidade de um aluno, grupo ou mesmo todos os alunos de um curso não executarem uma determinada actividade dentro da janela temporal prevista, o LMS poderia lançar uma actividade extra, destinada a corrigir a situação causada pelo incumprimento do plano do curso.

A execução de uma tal actividade abriria duas perspectivas de evolução no curso. Ou a repetição da actividade falhada ou o prosseguimento na actividade seguinte.

A figura 30 representa este cenário.

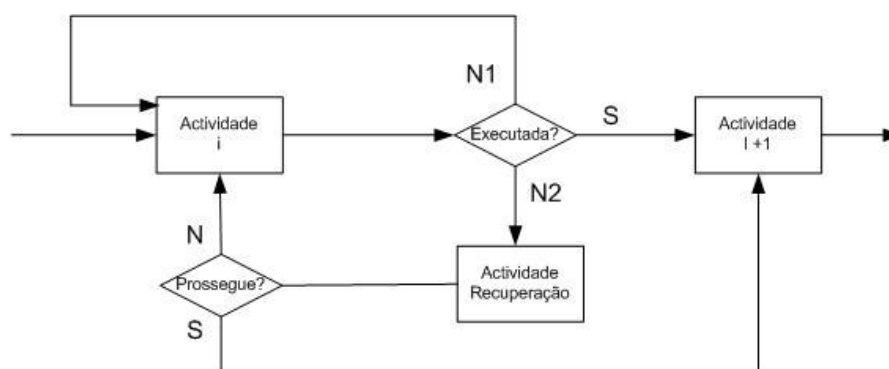


Figura 30 – Recuperação de actividade não executada.

Notar que o problema não se colocaria caso se estivesse em presença de um curso composto totalmente por actividades de execução aleatória, uma vez que a constatação de

que alguma actividade não havia sido cumprida só teria lugar quando o curso chegasse ao fim.

No caso de o curso ser composto por actividades calendarizadas, uma outra solução poderia passar pelo alargamento do prazo de execução da actividade para o limite temporal de execução da actividade seguinte. Neste caso o aluno teria que executar as duas actividades dentro desse novo espaço de tempo.

O problema, contudo, pode ser ainda mais complexo, já que o aluno em falta pode fazer parte de um grupo de trabalho e, assim, o seu incumprimento pode colocar em causa o desempenho de todo o grupo.

Uma solução mais global e eficaz consistiria na criação de uma nova instância do conjunto de actividades ainda por executar, com início na actividade em questão e apenas para o actor em causa.

#### 5.2.4 A componente notificações

Uma das principais motivações para a realização do presente trabalho residiu no facto de se pensar que uma componente de detecção automática de desvios na execução de actividades de um curso a distância, face ao plano do mesmo, potenciaria o aumento dos níveis de sucesso no processo de ensino/aprendizagem por parte dos participantes nesses cursos. Tal componente poderá ser utilizada com vantagens evidentes, permitindo a notificação de alunos, professores e equipas de suporte, em tempo útil, possibilitando desse modo intervenções correctivas para esses desvios.

##### 5.2.4.1 Tipos de notificações inicialmente previstos

Desde o início que foram considerados dois tipos de notificações - avisos e alarmes. Essas diferentes notificações foram representadas no modelo relativo à unidade atómica de um curso (cf. figura 2).

Relembre-se a importância dos avisos como forma de alertar os actores para a proximidade da execução de actividades. Por outro lado, os alarmes são justificados pelo que se disse no início desta secção.

A notificação por incumprimento de uma actividade calendarizada (segundo alarme) é apenas uma das situações em que se adivinha como interessante a utilização desse mecanismo. Uma outra é, sem dúvida, a que resulta da alteração de um plano de curso.

Deslocamento de actividades no tempo, inclusão ou remoção de actividades, ou até à redefinição dos conteúdos associados a uma actividade, todas são situações que com vantagens deverão ser acompanhadas de notificações dirigidas aos intervenientes relevantes no processo.

Os casos que se referiram apresentam contudo particularidades que os diferenciam entre si. No caso do incumprimento de uma actividade calendarizada, situação que tem servido para justificar a realização deste trabalho, aconselha-se um mecanismo de comunicação privado com o actor em falta.

Uma situação de alteração ao programa de um curso ou aos conteúdos a utilizar nas diversas actividades, poderia ser tratado de uma forma pública. Isto é, uma funcionalidade do tipo “quadro virtual de avisos” podia perfeitamente cumprir os mesmos objectivos de chamada de atenção para a alteração. Claro está que esta solução pressupõe o total empenhamento dos actores, já que só nesse caso será de esperar uma consulta constante das informações “colocadas” no “quadro virtual de avisos”. Colocando do lado dos diversos actores a responsabilidade de ir à procura da informação, estaria a ser esquecido o princípio básico que presidiu à realização deste trabalho. A solução por envio personalizado de uma mensagem, reduz ou anula a probabilidade de a informação não chegar ao(s) destinatário(s), pelo que se trata de uma solução potencialmente mais eficaz.

#### 5.2.4.2 Notificações do tipo informação

Como se poderá concluir a partir da observação da figura 2, inicialmente não estava previsto outro tipo de notificações que não fossem os “Avisos” e os “Alarmes” (primeiro e segundo). Porém, como resultado da reflexão que foi sendo efectuada e como consequência do próprio evoluir do trabalho, surgiu um terceiro tipo designado por “Informação”, cuja ocorrência se destina a informar os destinatários sobre uma qualquer situação decorrente do desempenho de um actor específico. Uma notificação deste tipo pode ser gerada em qualquer instante, no intervalo de tempo que vai desde o início da actividade até ao fim da mesma, não tendo, ao contrário dos “Avisos” e dos “Alarmes”, explícita ou implicitamente, definido um momento para a sua emissão.

Algumas situações que justificam a existência deste tipo de notificação podem ser identificadas. Considere-se o caso de um curso em ambiente não académico. É altamente provável que possa existir interesse por parte das chefias em receber informações acerca do empenhamento dos seus funcionários nas acções de formação em que estes participam.

Refira-se a título de exemplo um LMS proprietário da CISCO SYSTEMS<sup>49</sup> que notifica automaticamente os superiores hierárquicos sempre que um funcionário se inscreve voluntariamente num dos inúmeros cursos disponibilizados pela empresa. Da mesma forma, se o funcionário não conclui o curso, essa informação é enviada automaticamente às chefias.

Funcionalidades deste tipo podem até ser utilizadas internamente pelas organizações como meio complementar de decisão nos processos de avaliação dos funcionários.

Ainda no contexto deste tipo de notificações é possível identificar outros casos, observáveis em ambiente académico. Por exemplo, imagine-se a vantagem de o LMS informar um aluno de que o trabalho que ele submeteu já se encontra disponível com anotações e classificação. Está-se perante um caso em que a execução de uma actividade por parte do professor motivaria a emissão de uma notificação do tipo “Informação”, destinada ao aluno.

Repare-se que, ao contrário do que acontece com os alarmes, neste caso o gatilho para o envio da notificação não é a não execução de uma actividade dentro da janela temporal prevista, mas sim a execução com sucesso de uma actividade antes de atingido o limite para a execução da mesma.

A existir, esse tipo de notificação ocorre dentro da janela temporal prevista para a execução da actividade.

### 5.2.5 Os diferentes actores previstos

Iniciou-se este documento com a abordagem de aspectos dos LMS relacionados com as actividades, os eventos e as notificações, e com as relações que com eles são estabelecidas pelos diversos actores que se podem envolver num curso a distância via Web. Assim, faz sentido identificar e caracterizar os diferentes tipos de personagens que nesse contexto podem ser encontrados.

---

<sup>49</sup> <http://www.cisco.com/>

Alunos e professores são os dois tipos de actores que com maior probabilidade são identificados. Contudo, não devem ser esquecidos os elementos das equipas de suporte, importantíssimos para o bom funcionamento dos cursos. É a estes que se atribuem tarefas de apoio às dúvidas de utilização dos sistemas e de resolução de problemas técnicos, entre outros.

No decurso do presente trabalho identificou-se a necessidade de referenciar dois outros tipos de intervenientes, quanto ao papel que podem desempenhar no processo. Trata-se dos “Supervisores” e dos “Responsáveis”.

O Supervisor é a entidade que deve preocupar-se com a execução da actividade. É a ele que os actores a quem a actividade se destina devem recorrer se existirem dúvidas de interpretação relativas à mesma. Na maior parte dos casos coincidirá com o próprio professor, mas pode acontecer que o não seja.

De realçar o facto de não terem sido identificadas quaisquer actividades destinadas a ser executadas por este tipo de actor.

Por outro lado, a entidade “Responsável” assume-se como importante no processo, nomeadamente em ambientes não académicos. Trata-se de uma entidade que terá algum ascendente em termos organizacionais sobre um determinado aluno (que será de prever que seja um funcionário sob as ordens de tal entidade), podendo ser receptor de notificações do tipo “Informação” (cf. figura 31). O LMS deve possuir a capacidade de o notificar, em situações definidas aquando da arquitectura do curso, fazendo-lhe chegar informações que de alguma forma lhe podem ser úteis.

Numa situação não académica, é previsível a constituição de turmas cujos elementos podem ser originários de diferentes organizações, podendo verificar-se por esse motivo, a existência de “Responsáveis” também diferentes.

Tal como no caso anterior, também agora não se identificaram cenários em que existissem actividades dirigidas a este tipo de actores.

### 5.2.6 O modelo de gestão de actividades melhorado

O número de situações novas identificadas após a conclusão do projecto ODL Toolbox, justificou a introdução de alterações significativas no modelo identificado nesse projecto, bem como na inventariação de novas situações de interacção e na forma de lidar com as



mesmas. Foi necessário repensar processamentos já existentes e definir novos processos. Isto é, houve que alterar ou complementar as recomendações que deverão ser seguidas no desenvolvimento da camada de software, para que seja possível uma correcta manipulação dos metadados propostos neste trabalho.

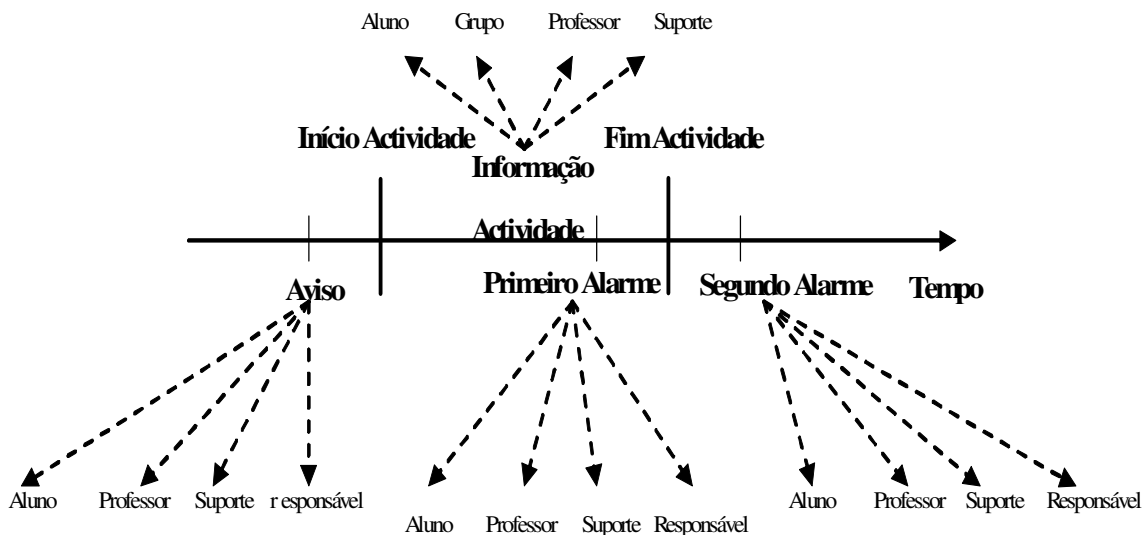


Figura 31 – Modelo melhorado da unidade atômica de gestão de actividades, em tempo real.

Entrando em linha de conta com todas as situações novas que se tem vindo a descrever, o modelo de unidade atômica de gestão de actividades em tempo real, representado na figura 2, evoluiu para a estrutura que a figura 31 apresenta, sendo de realçar no mesmo esquema o tipo de notificações que podem ser geradas em cada uma das quatro fases associadas a uma actividade. Essas combinações de fases e tipos de notificações aparecem sistematizadas na tabela 3.

Fases de uma actividade	Tipos de notificações
Antes do início da actividade	Aviso
Após o início da actividade	Informação
Antes de atingido o limite de execução da actividade	Primeiro alarme
Após ter sido atingido o limite de execução da actividade	Segundo alarme

Tabela 3- Fases associadas a uma actividade e tipos de notificações correspondentes.

### 5.2.7 Regras na definição de identificadores

No modelo de metadados proposto neste trabalho utiliza-se com frequência o elemento “Identificador”, substituído simplificadaamente pela forma “Id”. Embora potencialmente um utilizador pudesse criar identificadores sem qualquer outra restrição que não fosse a de já existirem, a indicação de uma regra para a sua criação justifica-se, como forma de uniformizar essa tarefa.

Um “Identificador” será definido formalmente de acordo com a gramática BNF (Backus-Naur Form) que a seguir se apresenta<sup>50</sup>:

```
Identificador ::= Id_aluno | Id_grupo | Id_professor | Id_curso |
                Id_executante | Id_responsável | Id_aviso | Id_alarme |
                Id_suporte | Id_informação | Id_actividade | Id_evento |
                Id_condição | Id_supervisor
Id_aluno ::= “ALU” {dígito_numérico}4
Id_grupo ::= “GRP” {dígito_numérico}4
Id_professor ::= “PRF” {dígito_numérico}4
Id_curso ::= “CUR” {dígito_numérico}4
Id_executante ::= “EXE” {dígito_numérico}4
Id_responsável ::= “RPS” {dígito_numérico}4
Id_aviso ::= “AVI” {dígito_numérico}4
Id_alarme ::= “ALM” {dígito_numérico}4
Id_suporte ::= “SPT” {dígito_numérico}4
Id_informação ::= “INF” {dígito_numérico}4
Id_actor ::= “ACT” {dígito_numérico}4
Id_evento ::= “EVT” {dígito_numérico}6
Id_condição ::= “CND” {dígito_numérico}4
Id_supervisor ::= “SUP” {dígito_numérico}4
Dígito_numérico ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

---

<sup>50</sup> A notação {dígito\_numérico}<sup>4</sup> significa a utilização de 4 elementos do tipo “dígito\_numérico”.

Por exemplo, o identificador ALU0025 identificaria o aluno número 25, enquanto que o identificador EVT000897 serviria para referir o registo inserido no repositório de eventos, com o número 897.

### 5.3 Funcionalidades da componente software

O que se descreveu até agora relativamente ao problema central deste trabalho, não deixa dúvidas quanto à necessidade da existência de uma estrutura de armazenamento de informação que terá que ser partilhada pelas camadas dos LMS já previstas no âmbito do projecto SCORM e por aquela que é proposta neste documento, relativa à gestão das actividades constituintes de um curso suportado por LMS.

Não se tratando de um problema tecnologicamente muito complicado, apresenta contudo, como se tem vindo a ver, muitas particularidades que exigem tratamento adequado por parte da camada de software que venha a ser desenvolvida com base nestas reflexões.

A solução passa pela utilização das entidades informacionais “Actividades” e “Eventos”, representando a primeira a própria estrutura de um curso e a segunda, a participação dos diversos actores no mesmo.

Neste contexto é importante a identificação de que componente dos LMS deve intervir, e de que forma o deve fazer, em cada situação concreta.

Um dos aspectos fulcrais neste trabalho reside na necessidade de serem armazenados registos correspondentes à execução das diversas actividades. Tais registos devem ser armazenados na entidade informacional “Eventos” e devem ser comparados com os planos dos cursos, materializados na entidade informacional “Actividades”. Assim, cada sessão de trabalho de um actor, para execução de uma actividade, deve ser analisada pelo LMS e apenas se a actividade tiver sido concluída com sucesso deve ser gerado o “evento” correspondente, mesmo que já tenha sido ultrapassado o limite temporal para a sua conclusão. Deste modo, a camada de gestão apenas lidará com informação útil constante do repositório “Eventos”. Ou seja, actividades que tenham sido iniciadas mas não tenham sido concluídas, não serão registadas nesse repositório.

No entanto, a opção poderia assentar na criação de um registo no repositório referido, sempre que uma actividade fosse iniciada por um actor. Tal registo apresentaria nessa

altura alguns atributos não preenchidos, dizendo respeito à conclusão da actividade. O LMS completaria posteriormente o preenchimento dos referidos atributos, quando e se a actividade fosse concluída.

Esta forma de tratar a situação apresentaria a vantagem de em termos de histórico ser possível identificar as situações de actividades que tivessem sido iniciadas e não terminadas.

Repare-se que esta abordagem coincide com o que se disse relativamente à “Execução de actividades ao longo de diversas sessões”.

De referir que a especificação SCORM já contempla o seguimento do estado de execução das actividades, pelo que não haverá qualquer dificuldade em realizar o registo do “evento” nas circunstâncias referidas.

A tarefa de emissão de notificações é da maior importância no modelo de gestão proposto. Tais notificações deverão ser geradas pela camada de gestão, segundo duas estratégias complementares:

- a) Por um lado, deve ser “percorrido” periodicamente o repositório das “Actividades” de um curso e, para cada uma delas, desde que tenha sido atingido o limite temporal para a sua execução, deve ser verificado no repositório dos “Eventos” se existem os correspondentes registos, relativos a todos os actores a quem essa actividade se destinava.

Nos casos em que tal se não verifique, deve ser solicitado o serviço de *messaging* do LMS, com a passagem dos parâmetros:

- Identificadores dos destinatários
- Mensagem do tipo alarme

Conhecido o identificador de cada um dos destinatários o sistema acederá ao meio preferencial de comunicação definido para os mesmos, ficando em condições de promover a sua notificação.

- b) Por outro lado, deverá ser verificado de forma também sistemática e periódica o repositório de eventos e identificados os casos em que as

actividades foram cumpridas dentro das janelas temporais previstas. Nesses casos e quando a actividade o indicar, o sistema de *messaging* do LMS deve ser solicitado, devendo a camada de gestão passar-lhe os parâmetros:

- Identificadores dos destinatários
- Mensagem do tipo informação

Também neste caso, conhecidos os identificadores dos destinatários, o sistema acederá à informação relativa ao meio preferencial de comunicação com os mesmos, e poderá levar por diante a tarefa de emitir as notificações necessárias.

O tratamento das situações relacionadas com actividades condicionadas é outro dos aspectos que os LMS devem cobrir.

No que à camada de gestão diz respeito, foram incluídos atributos nos elementos de metadados, capazes de suportar os procedimentos necessários.

Associou-se às actividades, atributos destinados a armazenar as condições que devem ser verificadas para que uma actividade possa ser executada, bem como um atributo capaz de informar se uma condição específica foi verificada ou não. Tal atributo deverá ser actualizado pelas outras camadas dos LMS com o valor verdadeiro ou falso conforme a condição se tenha ou não verificado.

Não deve perder-se de vista o facto de ser a essas camadas que cabe a tarefa de decidir quando é que uma actividade deve ou não ser apresentada para execução aos actores a quem se destina.

A inclusão dos elementos relativos às condições a verificar pode desta forma ser aproveitada pela camada de sequenciamento e navegação que, assim, poderá identificar as condições a testar antes da eventual apresentação da actividade.

Com base neste mecanismo, perante uma situação de não execução de uma actividade deste tipo, a camada de gestão emitirá um segundo alarme cujo conteúdo poderá incluir também a relação das condições não verificadas.

Por outro lado, e como já referido nas secções 5.2.2.5 (“Actividades condicionadas”) e 5.2.3.5 (“Execução condicionada de actividades”), a emissão do aviso de que está próximo

o lançamento de uma actividade deste tipo pode ser aproveitada para alertar o utilizador, caso alguma das condições não esteja verificada. Claro está que o lançamento da actividade, posteriormente, só será levado por diante se a camada de sequenciamento e navegação verificar que todas as condições que eram impostas para a apresentação da actividade, possuem um estado com valor igual a verdadeiro, no instante definido para o seu lançamento.

Uma outra funcionalidade que os LMS deverão apresentar relaciona-se com a actividade já descrita da constituição de grupos. O processo de gestão associado a este tipo de actividade, mais uma vez, envolve quer a camada de gestão, quer as outras camadas dos LMS.

Para cada uma destas duas componentes identificou-se um procedimento que se descreve de seguida.

- a) Ao atingir o instante final para execução da actividade, seleccionar cada aluno inscrito no curso e percorrer todos os grupos já iniciados (completos ou não), verificando se o aluno em causa existe em algum desses grupos. Em caso afirmativo, dar a actividade para esse aluno como executada.
- b) Para cada tentativa de inscrição num grupo, por parte de um aluno, ver se o mesmo já faz parte de algum grupo. Caso tal se verifique, o sistema deve impedir a nova inscrição, caso contrário devem ser verificadas as condições definidas para inclusão de um novo aluno num grupo. Por exemplo, os números máximo e mínimo de elementos por grupo e/ou o número máximo de grupos. Se as condições predefinidas não forem violadas, então deverá ser promovida a junção do aluno ao grupo e dar a actividade como executada, caso contrário, deve ser impedida a operação.

As actividades de grupo, nomeadamente aquelas que pressupõem a existência de uma estrutura de subactividades, cada uma das quais devendo ser executada por um elemento do grupo, são outra vertente que a camada *software* deve cobrir.

Para este caso, propõe-se uma solução baseada na apresentação de uma lista com as subactividades a serem executadas, podendo cada um dos elementos do grupo escolher

uma delas. Neste caso será necessário também ver implementado um mecanismo de controlo que vá retirando da lista das subactividades disponíveis e de uma forma dinâmica, aquelas que já tiverem sido atribuídas. Tal mecanismo não seria uma preocupação da camada de gestão mas sim das outras camadas dos LMS, nomeadamente a de sequenciamento e navegação, como já se referiu anteriormente. A camada de gestão continuaria apenas a ter que se preocupar com a determinação do estado de execução das diferentes subactividades dentro das janelas temporais previstas.

O LMS deverá também possuir mecanismos que evitem que um aluno possa escolher mais do que uma subactividade, como forma de garantir o envolvimento de todos os elementos do grupo na execução da actividade global.

O processo de identificação dos actores a quem se destina uma actividade também mereceu um tratamento cuidado, tendo sido possível isolar algumas características desejáveis, quer ao nível dos metadados, quer das funcionalidades da camada *software*.

Será conveniente considerar esta tarefa objecto de intervenções a dois tempos:

- a) Aquando da criação do curso, o autor deverá utilizar palavras reservadas para indicar a que classe de actores se destina a actividade.
- b) Numa segunda fase, coincidente com o início da entrada em funcionamento do curso, e após conhecidos (registados) todos os actores, o LMS deverá realizar automaticamente uma operação consistindo na associação de identificadores de actores concretos, às actividades.

As palavras reservadas consideradas são as seguintes:

- Todos
- Alunos
- Grupo
- Suporte
- Professor

Não pareceu que fosse de considerar o classificador “Aluno”, uma vez que será altamente improvável a existência de actividades dirigidas a um único aluno.

A utilização do classificador “Professor” deverá implicar o imediato preenchimento do identificador do mesmo, enquanto que ao associar-se a uma actividade o classificador “Grupo”, está-se a projectar para o momento imediatamente a seguir à criação de grupos a possibilidade de o LMS cumprir a segunda fase referida anteriormente, na alínea b).

## 5.4 Validação do modelo - Questões de competência

A figura 32 representa uma parte do esquema de base de dados que serviu de suporte à realização de testes sobre a adequabilidade do modelo ao problema subjacente ao trabalho descrito nesta tese. A validação foi realizada, fundamentalmente, através da verificação de um conjunto de questões de competência para as quais se procurou encontrar respostas que se pudessem obter a partir da manipulação das diversas tabelas que constituíam a base de dados.

Esta fase do trabalho teve início ainda durante o projecto ODL Toolbox referido anteriormente.

Internacionalmente, as ideias que motivaram a sua realização, já nessa altura foram bem recebidas. A prova-lo está a aceitação de alguns artigos apresentados em conferências internacionais, onde as reacções foram no sentido do reconhecimento do interesse do tema.

Referem-se a título de exemplo três artigos e respectivas conferências:

- Towards a Reference Framework for eLearning Management. Distance Learning Administration Conference (DLA2001), 2001, Callaway Gardens, Georgia, USA.
- Management of eLearning environments: some issues and research clues. ITEA Workshop, 2002, Innsbruck, Austria,
- Tracing dynamic behaviours in distributed learning communities. 16<sup>th</sup> Australian International Education Conference (AIEC), 2002, Hobart, Australia.



No modelo da figura 32, o plano do curso é implementado pela tabela “Actividades” enquanto que a tabela “Eventos” permite registar a informação associada à execução das diversas actividades planeadas.

Apresentam-se de seguida alguns exemplos de questões colocadas à base de dados e da forma como é possível obter respostas para as mesmas.

Questões:

1. Que utilizadores se encontram inscritos no curso com o estatuto de aluno?
2. Há necessidade de emitir algum aviso neste momento? Com que conteúdo?
3. Será necessário emitir algum alarme neste momento? Destinado a quem?
4. Qual a próxima actividade prevista no plano do curso?

Como obter as respostas às questões colocadas:

1. Uma interrogação à base de dados, utilizando a tabela actores e filtrando os registos pelo atributo Tipo = "aluno" permitiria responder à questão.
2. É possível obter todos os registos da tabela Actividades em que se verifique a condição:  $Data\_actual + conteúdo\ do\ atributo\ dias\_aviso \geq conteúdo\ do\ atributo\ Dia\_início$ . Para cada um desses registos, identifica-se na tabela Avisos o aviso cujo conteúdo do atributo Id\_actividade coincide com o conteúdo do atributo Id\_actividade da tabela Actividades.
3. Dada uma actividade cujos conteúdos dos atributos Dia\_fim e Hora\_fim sejam inferiores ou iguais ao dia e hora actuais, para cada Id\_actor na tabela Actores, com estatuto pertencente ao conjunto dos indicados pelo conteúdo do atributo Destinatários, da tabela Actividades, ver se existe na tabela Eventos um evento com  $Id\_evento = Id\_actividade$  da tabela Actividades e com  $Id\_actor = Id\_actor$  da tabela Actores, tal que  $Data\_início$  e  $Hora\_início$  da tabela Eventos seja menor ou igual ao  $Dia\_fim$  e  $Hora\_fim$  da tabela Actividades. Caso não exista, emitir o alarme cujo  $Id\_actividade = Id\_actividade$  da tabela Actividades, com destino para o utilizador com  $Id\_actor$ .
4. Encontrar na tabela Actividades o primeiro registo cujos conteúdos dos atributos  $Data\_início$  e  $Hora\_início$  sejam  $\geq$  Data e Hora actuais.

O exercício de que se acabou de apresentar um excerto permitiu concluir que o modelo era potencialmente adequado ao tratamento do problema que motivou o trabalho descrito nesta tese. Tal conclusão pareceu aceitável, dado que se conseguia obter respostas para todas as questões colocadas à base de dados.

Assim sendo, o desenvolvimento de uma camada de software que manipulasse a base de dados apresentada, dotada das funcionalidades necessárias, poderia constituir um protótipo de um LMS verificando as características que foram sendo mencionadas no texto presente.

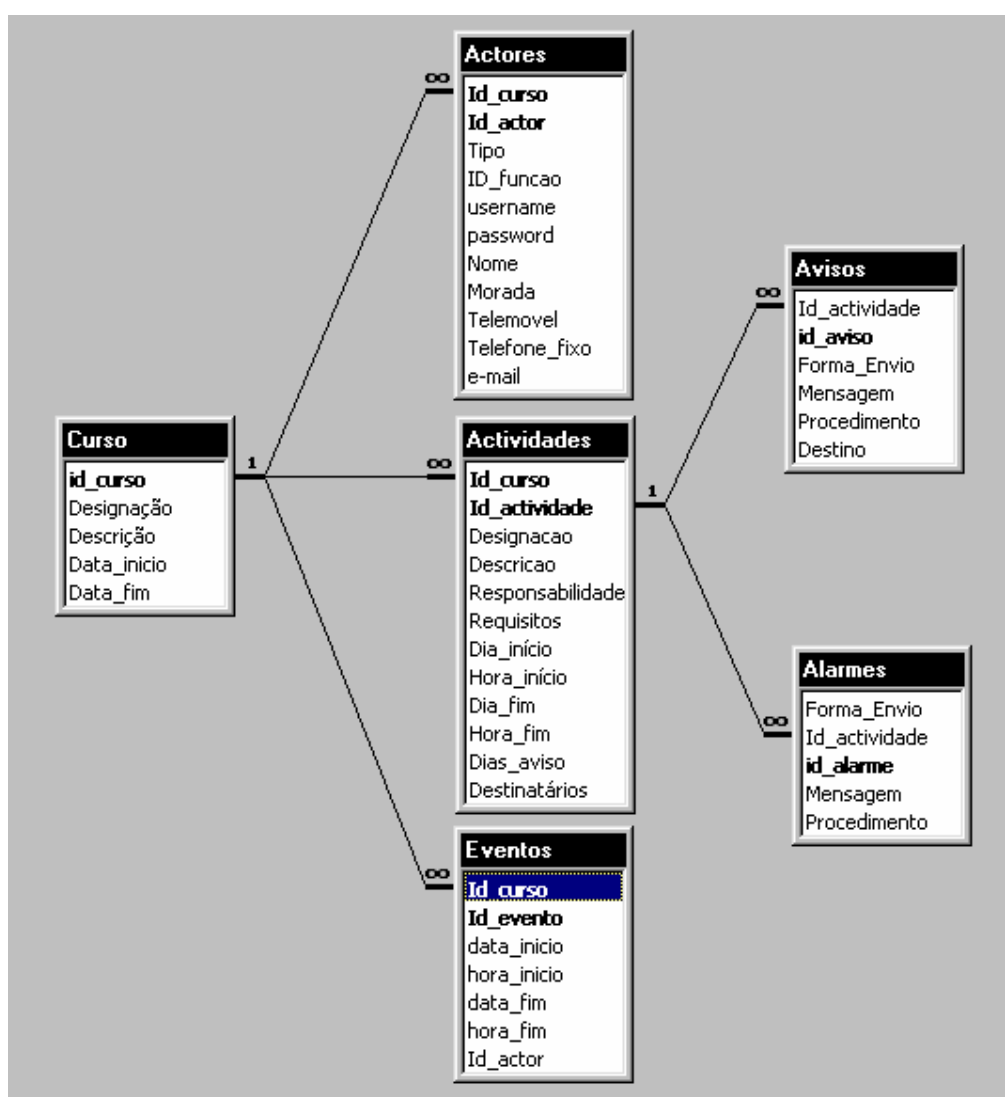


Figura 32 – Vista parcial do esquema de base de dados de suporte a um LMS.

## 5.5 Arquitectura proposta

A arquitectura que se propõe para um LMS é apresentada na figura 33 e baseia-se na estrutura proposta pelo projecto SCORM estendida com:

- A camada de gestão operacional dos cursos;
- Um repositório de “Eventos” correspondendo ao registo das interacções dos diversos actores com o curso e o LMS;
- A adaptação do repositório onde são registadas as “Actividades” constituintes de um curso;
- A adaptação e introdução de novos métodos na API do SCORM RTE;
- Um mecanismo de solicitação do sistema de *messaging* do LMS, para implementar o processo de notificações.

A maioria das plataformas de *eLearning* incluem mecanismos de comunicação entre os diferentes actores. Algumas das actividades que podem ser encontradas num curso podem basear-se na comunicação síncrona ou assíncrona entre dois ou mais participantes nesse curso, através de sistemas de *chat*, de correio electrónico ou de *fora* de discussão.

Dadas as características da comunicação entre os diversos actores e entre estes e os cursos, via funcionalidades de comunicação e interface dos LMS, é aceitável considerar as reacções da camada de gestão com algum atraso temporal, como sendo reacções em tempo real. Isto é, esse atraso não distorce significativamente o princípio da reacção em tempo real do sistema, a situações anormais de interacção dos utilizadores com os cursos ou outras funcionalidades disponibilizadas pelos LMS.

Por exemplo, dado que a emissão automática de alarmes em situações de não execução das actividades previstas no plano do curso é uma das funcionalidades que justifica este trabalho, a camada de gestão do LMS fará leituras sistemáticas dos repositórios de “Actividades” e “Eventos” para decidir quando deve ser gerada uma notificação desse tipo. A figura 33 representa o relacionamento da camada de gestão com as outras camadas do LMS, bem como as interacções dos utilizadores com o sistema.

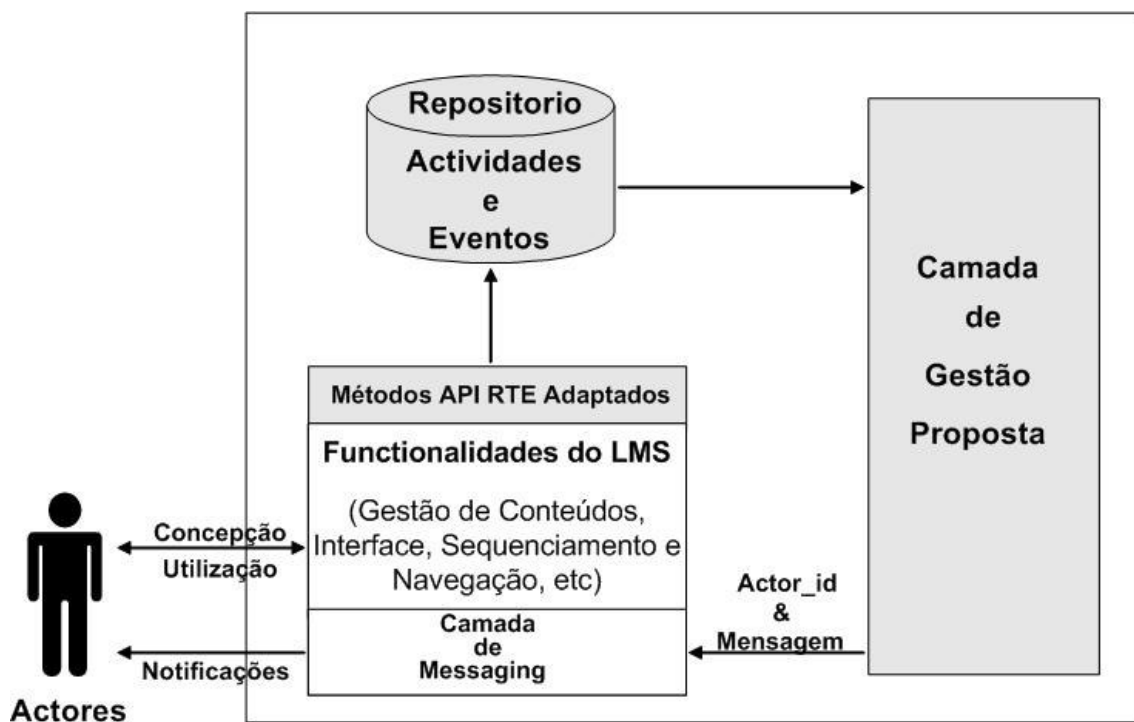


Figura 33 – Arquitectura de relacionamento da camada de gestão com o restante LMS e das interações dos diferentes actores com o sistema global.

A leitura que deve ser feita do esquema da figura 33 é a seguinte:

1. Os autores dos cursos interagirão com a plataforma no sentido de construir os cursos, registando entre outra informação, aquela que implementa a estrutura do próprio curso, isto é, as actividades a realizar no âmbito do mesmo.
2. Posteriormente, os actores para os quais existam “actividades” definidas, interagirão com a plataforma no sentido de as executarem e, dessa interacção, resultará o registo dos diversos “eventos”.
3. Os diversos actores poderão também utilizar os mecanismos de comunicação síncrona e/ou assíncrona fora do âmbito da execução de uma actividade, para comunicarem informalmente entre si.
4. Permanentemente, a camada de gestão do LMS consultará os repositórios de actividades e eventos para identificar situações que justifiquem a emissão de notificações do tipo primeiro alarme ou segundo alarme. Existindo casos que

justifiquem essa emissão, a camada de gestão solicitará a camada de *messaging* da plataforma, passando-lhe a informação necessária, constituída por conjuntos de pares com os seguintes componentes:

- identificadores dos destinatários;
- Identificador da mensagem.

5. Finalmente, o LMS utilizando as suas funcionalidades de *messaging*, após identificar o meio preferencial de comunicação de cada destinatário, enviará as notificações para os destinos apropriados, conforme a informação recebida da camada de gestão, ou criará as condições para que essas notificações sejam enviadas por formato não electrónico, se for esse o caso.

De referir que pode existir mais do que um destinatário para uma notificação, nomeadamente quando se tratar de enviar mensagens para um grupo.

Mesmo o caso de destinatários de diferentes tipos, eventualmente recebendo mensagens diferentes, está bem suportado pelo modelo global de metadados de gestão proposto neste trabalho, como se poderá depreender da observação da estrutura de pares informacionais apresentada.

### 5.5.1 Metadados de gestão

Uma das virtudes normalmente associadas à utilização de metadados, tem que ver com o facto de estes acrescentarem informação à informação, por um lado, e de permitirem a realização de operações de pesquisa e recuperação de informação mais eficientes, por outro.

No caso presente, a inclusão desta secção na tese justifica-se pelo facto de o termo referir potencialidades que no contexto do trabalho realizado assumem uma importância vital. De facto, dada a intenção de tentar propor uma componente que fosse integrável no todo que constitui um LMS construído à luz das recomendações SCORM, essa componente teria que ser compatível com as restantes camadas propostas pelo projecto referido, de forma a garantir com essa integração, um dos objectivos mais perseguidos

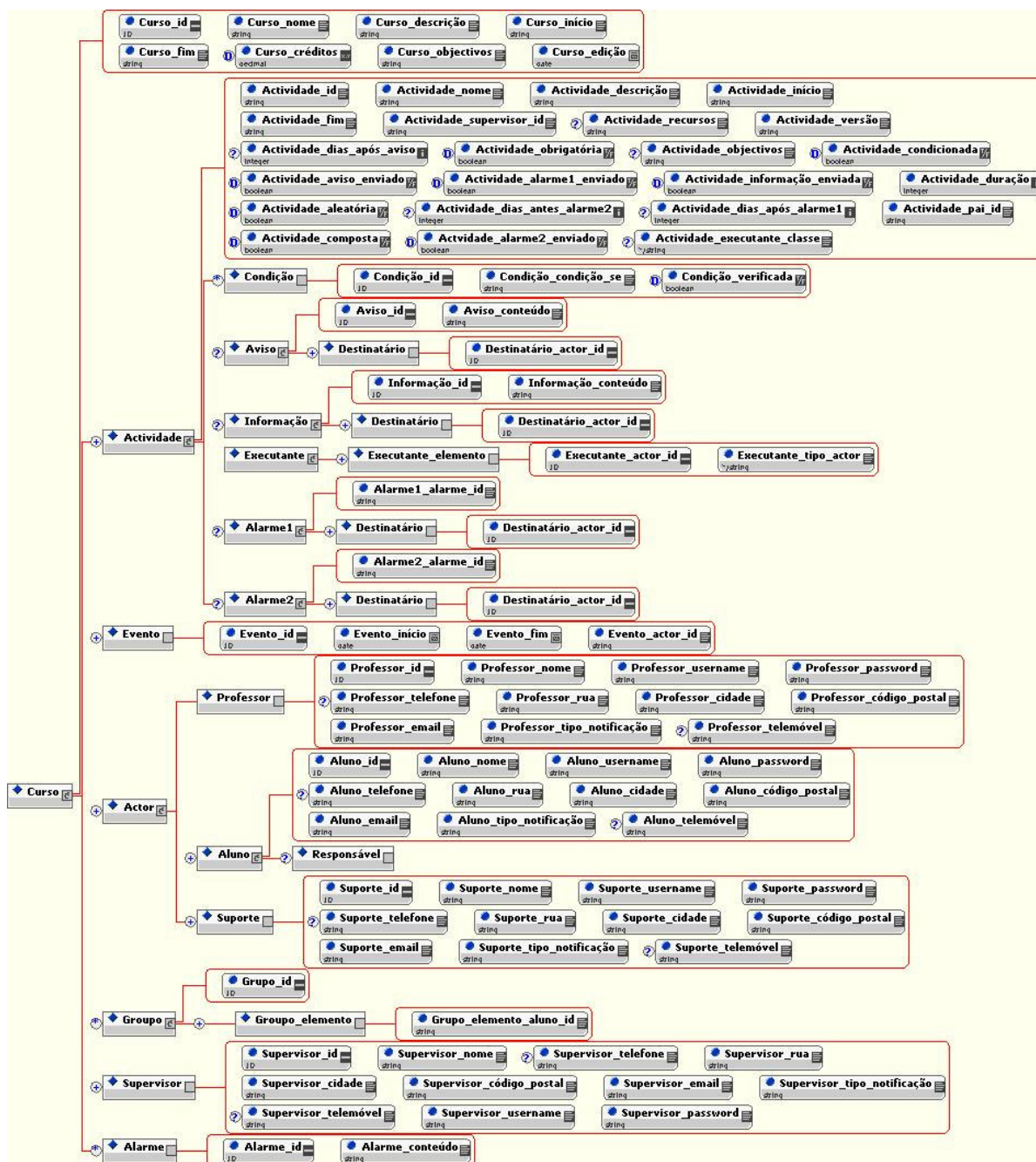


Figura 34 – Representação gráfica do esquema XML para a camada de gestão, obtida via Turbo XML da TIBCO.

pelos projectos de normalização que se têm vindo a referir – a interoperabilidade dos sistemas.

A figura 34 apresenta o esquema XML para a camada de gestão proposta, a partir da utilização da ferramenta Turbo XML da TIBCO<sup>51</sup>.

Na verdade, a versão gráfica do esquema XML apresentado na figura 34 não se limita a representar o conjunto dos elementos pertencentes à camada de gestão de actividades dos LMS. O que se pode observar no esquema referido é a estrutura de elementos de metadados de um LMS completo, incluindo a vertente das interacções dos diferentes actores com o mesmo.

Far-se-á seguidamente uma apresentação parcial de algumas das componentes mais importantes da estrutura global apresentada na figura 34, descrevendo-se sempre que achado necessário, a função de cada um dos elementos de metadados utilizados, bem como a forma como se propõe que os mesmos sejam preenchidos nas várias fases que vão desde a construção de um curso, até à execução das actividades que o constituem.

De referir apenas que cada elemento de metadados do esquema é definido à custa de atributos e/ou outros elementos.

A leitura deste conjunto de diagramas impõe que se estabeleça o significado da simbologia utilizada na ligação dos ramos da estrutura. Assim, o leitor deve ter presente que:

- “+” - O elemento pode existir uma ou mais vezes.
- “\*” - O elemento pode existir zero ou mais vezes.
- “?” - O elemento é opcional.
- “D” - O atributo tem associado um valor por defeito.
- A inexistência de qualquer símbolo significa que o elemento é de utilização obrigatória e que ocorre uma e uma só vez.

A figura 35 apresenta uma versão mais sintética dos elementos que compõem um curso. Da sua observação resulta claro que um curso, para além dos seus atributos caracterizadores (Curso\_id, Curso\_nome, Curso\_descrição, Curso\_início, Curso\_fim, Curso\_créditos, Curso\_objectivos e Curso\_edição), é composto por um conjunto de

---

<sup>51</sup> <http://www.tibco.com/>

elementos que podem ocorrer com diferentes multiplicidades, de acordo com a simbologia que acabou de ser apresentada.

Notar que a caracterização completa de todos os elementos e atributos será apresentada mais adiante na tabela 4.

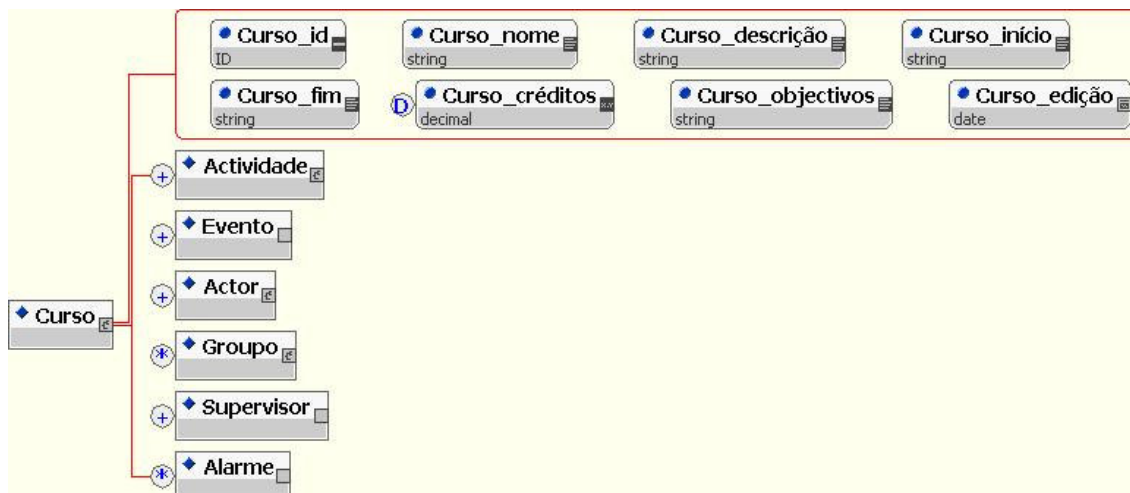


Figura 35 – Representação gráfica resumida do esquema XML relativo a um Curso.

De acordo com os pressupostos discutidos em todo o capítulo 5 desta tese, a figura 36 apresenta o detalhe de uma “Actividade”.

De realçar o facto de se ter optado por não considerar uma classe “Notificação”, dizendo respeito a Avisos, Informações e Alarmes. Esta opção prende-se com o facto de cada um desses tipos de notificações possuir atributos específicos, que as tornam diferentes entre si.

Recordando a função da entidade informacional “Evento”, esta implementa o repositório de informação relativo às interações dos utilizadores com os cursos, no que toca à execução das actividades programadas.

As restantes camadas de um LMS devem registar neste repositório as ocorrências correspondentes à conclusão das actividades, sendo com base em tal informação que a camada de gestão, por comparação com o plano do curso, consegue determinar se existe necessidade da emissão de qualquer notificação do tipo “Alarme” ou “Informação”.



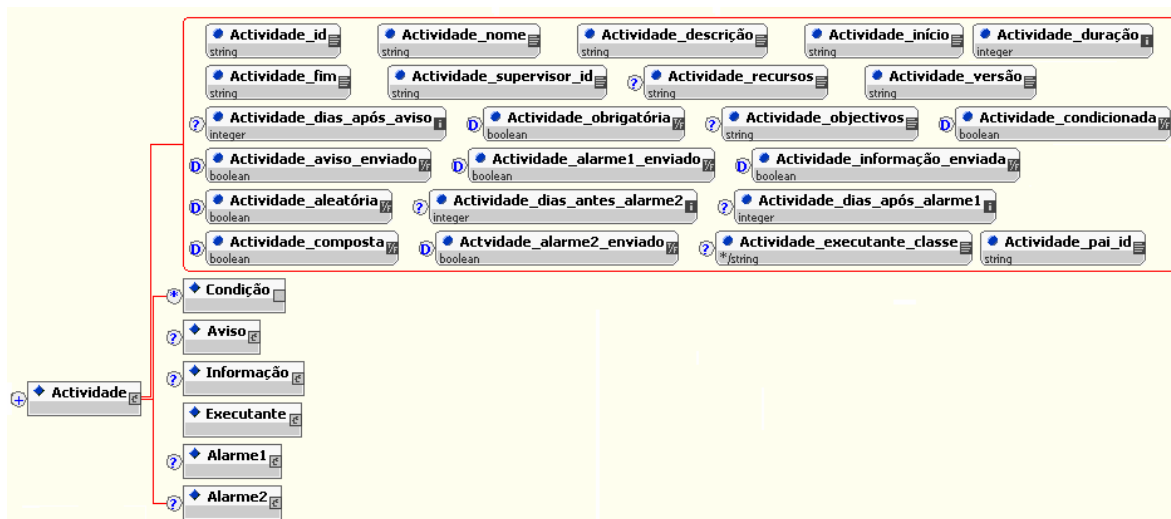


Figura 36 – Subesquema XML para a componente Actividade.

A figura 37 representa este elemento, sendo claro que com base nos atributos “Evento\_actor\_id”, “Evento\_id” e “Evento\_fim”, se torna possível determinar que actor terminou que actividade e se o fez dentro da janela temporal prevista.

É importante relembrar as duas opções apresentadas para a criação de uma instância desta classe. Ou se promove a sua criação apenas quando uma actividade for concluída ou, alternativamente, o sistema criará uma instância da classe quando se verificar o início da actividade, em que o atributo “Evento\_fim” ficará sem conteúdo. Esse atributo poderá ser posteriormente preenchido, caso a actividade seja concluída, ou ficará com o referido atributo sem conteúdo, denotando que a actividade não foi executada.



Figura 37 – Subesquema XML para a componente Evento.

A classe “Actor” é desde o princípio aquela que menos dúvidas introduziu na execução do presente trabalho. Com efeito, para que haja um curso em funcionamento é preciso que em algum instante alguém o tenha arquitectado e “montado” (Professor), é preciso que existam alunos interessados na frequência desse curso e, desejavelmente, deverá existir

uma equipa de suporte capaz de apoiar os diferentes actores durante a utilização da plataforma tecnológica (LMS), garantindo também a operacionalidade constante dos recursos tecnológicos necessários.

A figura 38 apresenta os elementos de metadados que se acabou de referir, sendo apenas de realçar a particularidade da associação opcional do “Responsável” a um “Aluno”, permitindo cobrir as situações descritas na secção 5.2.5 deste documento.

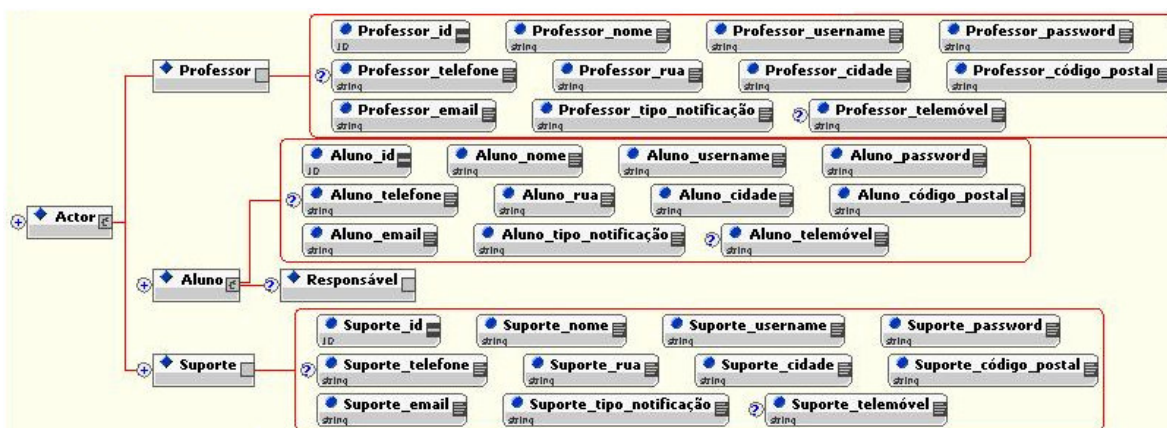


Figura 38 – Subesquema XML para a componente Actor.

Por fim, mas com o realce que lhe é conferido pelo facto de poder ser uma contribuição para o projecto SCORM, apresenta-se na figura 39 a classe “Grupo”. Note-se que o projecto SCORM considera a entidade “aluno” apenas isoladamente, não estando previstas portanto, actividades de grupo.

A classe “Grupo”, como se pode verificar, apenas exige que se forneça um identificador pelo qual o grupo passará a ser reconhecido no sistema e, um conjunto com pelo menos um elemento do tipo “aluno”<sup>52</sup>.

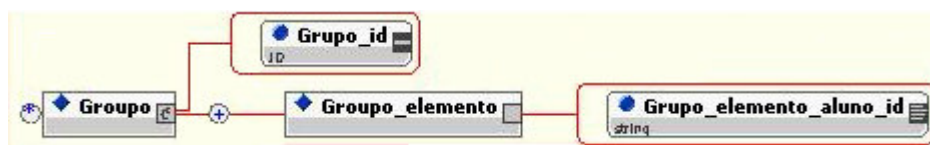


Figura 39 – Subesquema XML para a componente Grupo.

<sup>52</sup> Apenas limitações da ferramenta utilizada impediram que se representasse um grupo como tendo que possuir pelo menos dois elementos.

Apresenta-se seguidamente a hierarquia de classes correspondente ao modelo proposto para a camada de gestão de actividades, bem como a descrição de todos os elementos de metadados correspondentes.

De referir que a ordem pela qual os diversos elementos são apresentados é irrelevante.

## Hierarquia de classes

---

- :THING
    - Curso
    - Actividade
    - Evento
    - Actor
      - Aluno
      - Suporte
      - Professor
    - Notificação
      - Alarme1
      - Alarme2
      - Informação
      - Aviso
    - Condição
    - Grupo
    - Responsável
    - Supervisor
- 

### Classe Curso

**Classe Concreta Estende**  
:THING

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

**Documentação da Classe:**

Esta classe destina-se a caracterizar cada um dos cursos hospedados no LMS.

<b>Atributo</b>	<b>Documentação</b>	<b>Tipo</b>	<b>Valores/Classes Permitidas</b>	<b>Cardinalidade</b>	<b>Valor por Defeito</b>
Curso_id	Este atributo deve ser preenchido com o código identificador do curso. Tal código deve ser único.	String		1:1	
Curso_nome	Este atributo deve ser preenchido com o nome do curso. Este é um atributo de preenchimento obrigatório.	String		1:1	
Curso_descrição	Este atributo deve ser preenchido com uma descrição completa do curso. Considera-se este atributo de preenchimento obrigatório.	String		1:1	
Curso_início	Este atributo deve ser preenchido com a data de início do curso. O conteúdo deste atributo terá que ser anterior ao do atributo Curso_fim.	String	Data_hora em formato ISO8601:2004	1:1	
Curso_fim	Este atributo deve ser preenchido com a data de fim do curso. Esta data terá que ser posterior à que constitui o conteúdo do atributo Curso_início.	String	Data_hora em formato ISO8601:2004	1:1	
Curso_créditos	Este atributo deve ser preenchido com o número de créditos do curso. Se o curso não conferir qualquer número de créditos, o atributo deve ser mantido com o valor por defeito – zero.	Float		1:1	0.0
Curso_objectivos	Este atributo deve ser utilizado para descrever os objectivos do curso. O seu preenchimento é facultativo.	String		1:1	
Curso_edição	Este atributo deve ser preenchido com o conteúdo do atributo Curso_início. Assume-se que não se inicia o mesmo curso mais do que uma vez na mesma data.	String	Curso_início	1:1	

## Classe Actividade

Classe Concreta Estende  
:THING

Instâncias Directas:  
Nenhuma

Subclasses Directas:  
Nenhuma

### Documentação da Classe:

Esta classe representa a estrutura dos cursos. Considera-se um curso como sendo um conjunto de actividades planeadas.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Actividade_id	Este atributo deve ser preenchido com o identificador da actividade se não existir uma decomposição dessa actividade em subactividades. No caso em que tal decomposição exista, o atributo deve ser preenchido com o identificador da actividade pai, caso se trate de uma subactividade.	String		1:1	
Actividade_nome	Este atributo deve ser preenchido com o nome da actividade.	String		1:1	
Actividade_descricao	Este atributo deve ser preenchido com a descrição completa da actividade.	String		1:1	
Actividade_inicio	Este atributo deve ser preenchido com a data e a hora de início da actividade. Nos casos em que a execução da actividade é aleatória, o atributo deve ser preenchido pelo LMS com os valores de data e hora do sistema, que se	String	Data_hora em formato ISO8601:2004	1:1	

	verifiquem no momento de início de execução da mesma.				
Actividade_fim	Este atributo deve ser preenchido com a data e a hora previstas para o fim da execução da actividade. Nos casos em que a actividade seja de execução aleatória, o atributo deverá ser preenchido pelo LMS, com base no conteúdo dos atributos Actividade_início e Actividade_duração.	String	Data_hora em formato ISO8601:2004	1:1	
Actividade_supervisor_id	Este atributo deve ser preenchido com o identificador da pessoa que é responsável pelo controlo da execução da actividade. É a esta pessoa que os diversos actores a quem a actividade se destina, devem recorrer em caso de dúvidas sobre a mesma.	String	Professor_id	1:1	
Actividade_recurros	Este atributo descreve os recursos necessários à execução da actividade. Tais recursos podem ser tecnológicos, humanos, materiais de suporte, etc.	String		0:1	
Actividade_versão	Este atributo deve ser preenchido com o identificador da versão da actividade. Dado que podem existir várias evoluções da mesma actividade, é necessário saber que versão está a ser utilizada numa determinada edição do curso.	String		1:1	
Actividade_dias_antes_alarme2	Este atributo deve ser preenchido com o número de dias que deve ser deixado passar após o fim da actividade, para que seja emitido eventualmente o segundo alarme (no caso de a actividade não ter sido executada dentro da janela temporal definida, por algum dos destinatários previstos). Uma actividade	Inteiro		0:1	

	<p>pode não ter associado um segundo alarme! O atributo pode, portanto, ser deixado sem qualquer valor.</p> <p>Se o segundo alarme dever ser emitido imediatamente a seguir ao fim da actividade, indicado pelo conteúdo do atributo <code>Actividade_fim</code>, no caso em que a mesma não tenha sido executada por algum dos destinatários previstos, então o atributo deve ser preenchido com o valor zero.</p>				
<code>Actividade_dias_após_alarme1</code>	<p>Este atributo deve ser preenchido com o número de dias de antecedência com que o primeiro alarme deve ser emitido, relativamente à data de fim da actividade, se esta não estiver em curso e se não estiver concluída. Uma actividade pode não ter associado um primeiro alarme! O atributo pode, portanto, ser deixado sem qualquer valor.</p>	Inteiro		0:1	
<code>Actividade_dias_após_aviso</code>	<p>Este atributo deve ser preenchido com o número de dias de antecedência com que o aviso de proximidade de uma actividade deve ser enviado aos destinatários dessa actividade. Uma vez que podem existir actividades de execução aleatória, este atributo pode ser deixado sem conteúdo.</p>	Inteiro		0:1	
<code>Actividade_aviso_id</code>	<p>Este atributo deve ser preenchido com um identificador que deve ser único e ser uma instância da classe <code>Avisos</code>. Considera-se a possibilidade de uma</p>	String		0:1	

	actividade não ter o seu início predefinido (actividades de execução aleatória). Nesses casos este atributo não poderá ser preenchido previamente.				
Actividade_informação_id	Este atributo deve ser preenchido com um identificador de informação válido. Esse identificador terá que ser uma instância da classe Informação.	String		0:1	
Actividade_alarme1_id	Este atributo deve ser preenchido com um identificador de um alarme que deve ser único e uma instância da classe Alarmes, no caso de existir um primeiro alarme definido para a actividade. Uma actividade pode não ter associado um primeiro alarme, portanto, este atributo pode ser deixado sem conteúdo.	String		0:1	
Actividade_alarme2_id	Este atributo deve ser preenchido com um identificador de um alarme que deve ser único e uma instância da classe Alarmes, no caso de existir um segundo alarme definido para a actividade. Uma actividade pode não ter associado um segundo alarme, portanto, este atributo pode ser deixado sem conteúdo.	String		0:1	
Actividade_obrigatória	Este atributo deve ser preenchido com o valor Verdadeiro ou Falso, dependendo de se a actividade é de execução obrigatória ou não.	Lógico		1:1	Verda-deiro
Actividade_composta	Este atributo deve ser preenchido com o valor Verdadeiro ou Falso, dependendo de se a actividade tem ou não subactividades.	Lógico		1:1	falso
Actividade_objectivos	Este atributo deve ser preenchido com a	String		1:1	



	descrição dos objectivos previstos para a actividade.				
Actividade_condicionada	Este atributo indica se a actividade é ou não de execução condicionada. Isto é, se existem pré-condições para a execução da mesma.	Lógico		1:1	falso
Actividade_aviso_enviado	Este atributo deve ser preenchido com o valor Verdadeiro no caso em que o aviso tenha sido enviado aos elementos constantes do atributo Aviso_destino.	Lógico		1:1	falso
Actividade_alarme1_enviado	Este atributo deve ser preenchido com o valor Verdadeiro no caso em que o alarme1 tenha sido enviado aos elementos constantes do atributo “alarme1_destino”.	Lógico		1:1	falso
Actividade_alarme2_enviado	Este atributo deve ser preenchido com o valor Verdadeiro no caso em que o alarme2 tenha sido enviado aos elementos constantes do atributo “alarme2_destino”.	Lógico		1:1	falso
Actividade_informação_enviada	Este atributo deve ser preenchido com o valor Verdadeiro se uma notificação do tipo Informação tiver sido enviada aos destinatários previstos, constantes do atributo Informação_destino.	Lógico		1:1	falso
Actividade_pai_id	Este atributo deve ser preenchido com o identificador da actividade se a mesma não for decomposta em subactividades. Caso contrário, o atributo deve ser preenchido com o identificador da actividade da qual esta deriva (actividade pai).	String		1:1	
Actividade_aleatória	Este atributo deve ser preenchido com um valor lógico Verdadeiro ou	Lógico		1:1	falso

	Falso, indicando se a actividade tem ou não um instante de início definido.				
Actividade_executante_classe	Este atributo deve ser preenchido com um dos valores permitidos.	Símbolo	Todos, Alunos, Grupo, Suporte, Professor	0:1	
Actividade_duração	Este atributo deve ser preenchido com o número de dias definido para a execução da actividade.	Inteiro		1:1	

## Classe Evento

**Classe Concreta Estende**  
:THING

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

### Documentação da Classe:

Nesta classe é possível registar toda a informação relativa às interações entre os diversos actores e os cursos, de acordo com as actividades planeadas. As plataformas podem comparar a informação registada nesta classe com a informação constante da classe “Actividades”. Dessa forma será possível decidir quando gerar e enviar notificações do tipo “alarme1” ou “alarme2”.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Evento_id	Este atributo deve ser preenchido com um dos identificadores das actividades do curso, significando que essa actividade foi executada pelo actor cujo identificador constará do atributo Evento_actor_id.	String		1:1	
Evento_início	Este atributo deverá ser preenchido pelo LMS com a data e hora correspondente ao início da execução da actividade por parte do actor cujo identificador se encontra	String	Data_hora em formato ISO8601:2004	1:1	

	no atributo Evento_actor_id.				
Evento_fim	Este atributo deverá ser preenchido pelo LMS com a data e hora correspondente ao fim da execução da actividade por parte do actor cujo identificador se encontra no atributo Evento_actor_id.	String	Data_hora em formato ISO8601:2004	1:1	
Evento_actor_id	Este atributo deve ser preenchido com um identificador válido de um actor a participar no curso (uma instância da classe Actor).	String		1:1	

## Classe Actor

**Classe Abstracta Estende:**  
:THING

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**

1. Professor
2. Aluno
3. Suporte

**Documentação da Classe:**

Assume-se que todo o actor deve possuir um endereço de correio electrónico.

## Classe Aluno

**Classe Concreta Estende**  
Actor

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

### Documentação da Classe:

Esta classe representa os actores que desempenham o papel de alunos.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Actor_nome	Este atributo deve ser preenchido obrigatoriamente com o nome do actor.	String		1:1	
Actor_username	Este atributo deve ser preenchido obrigatoriamente com o "username" do actor.	String		1:1	
Actor_password	Este atributo deve ser preenchido obrigatoriamente com a "password" do actor. Esta, deverá ser única.	String		1:1	
Actor_telefone	Este atributo deve ser preenchido com o número de telefone do actor, caso exista.	String		0:1	
Actor_rua	Este atributo deve ser preenchido com o nome da rua e o número de polícia da morada do actor.	String		1:1	
Actor_cidade	Este atributo deve ser preenchido com o nome da cidade da morada do actor.	String		1:1	
Actor_código_postal	Este atributo deve ser preenchido com o código postal da morada do actor.	String		1:1	
Actor_email	Este atributo deve ser preenchido com o endereço de correio electrónico do actor. Assume-se que nenhum actor pode participar num curso sem que possua um endereço de correio electrónico.	String		1:1	
Actor_notificação_tipo	Este atributo deve ser preenchido com a	Símbolo	SMS, eMail, Telefone,	1:1	

	identificação do modo preferencial de comunicação com o actor. Esse modo terá que ser um dos valores permitidos.		Telefone_móvel, Carta.		
Actor_ telefone_móvel	Este atributo deve ser preenchido com o número de telefone móvel do actor, caso exista.	String		0:1	
Aluno_ responsável_id	Em contextos não académicos é possível considerar a existência de uma pessoa que seja responsável pelo aluno. Nesses casos, o atributo deve ser preenchido com um identificador válido de um responsável. Esse identificador deverá ser uma instância da classe Responsável.	String		0:1	
Aluno_id	Este atributo deve ser preenchido com um identificador que terá que ser único e uma instância da classe Aluno.	String		1:1	

## Classe Suporte

### Classe Concreta Estende

Actor

### Instâncias Directas:

Nenhuma

### Subclasses Directas:

Nenhuma

### Documentação da Classe:

Esta classe representa os actores que têm como missão o suporte aos outros actores durante a sua interacção com os cursos e as plataformas.

Assume-se que não há necessidade de individualizar cada elemento das equipas de suporte. Assim, actividades de suporte são associadas à função Suporte como um todo.

Da mesma forma as notificações que tenham como destino os elementos dessas equipas, possuirão como identificador de destino apenas “Suporte”, não sendo relevante que elemento concreto dessas equipas deve receber a notificação.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Actor_nome	Este atributo deve ser preenchido com o nome do actor.	String		1:1	
Actor_username	Este atributo deve ser preenchido com o “username” do actor.	String		1:1	
Actor_password	Este atributo deve ser preenchido com a “password” do actor. Essa “password” deve ser única.	String		1:1	
Actor_telefone	Este atributo deve ser preenchido com o número de telefone do actor, caso exista.	String		0:1	
Actor_rua	Este atributo deve ser preenchido com o nome da rua e o número de polícia da morada do actor.	String		1:1	
Actor_cidade	Este atributo deve ser preenchido com o nome da cidade da morada do actor.	String		1:1	
Actor_código_postal	Este atributo deve ser preenchido com o código postal da morada do actor.	String		1:1	
Actor_email	Este atributo deve ser preenchido com o endereço de correio electrónico do actor. Assume-se que nenhum actor pode participar num curso sem que possua um endereço de correio electrónico.	String		1:1	
Actor_notificação_tipo	Este atributo deve ser preenchido com a identificação do modo	Símbolo	SMS, eMail, Telefone, Telefone_móvel,	1:1	

	preferencial de comunicação com o actor. Esse modo terá que ser um dos valores permitidos.		Carta.		
Actor_telefone_móvel	Este atributo deve ser preenchido com o número de telefone móvel do actor, caso exista.	String		0:1	
Suporte_id	Este atributo deve ser preenchido com um identificador que terá que ser único e uma instância da classe Suporte.	String		1:1	

## Classe Professor

### Classe Concreta Estende

Actor

### Instâncias Directas:

Nenhuma

### Subclasses Directas:

Nenhuma

### Documentação da Classe:

Esta classe representa os actores que desempenham o papel de professores.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Actor_nome	Este atributo deve ser preenchido com o nome do actor.	String		1:1	
Actor_username	Este atributo deve ser preenchido com o "username" do actor.	String		1:1	
Actor_password	Este atributo deve ser preenchido com a "password" do actor.	String		1:1	

	Essa “password” deve ser única.				
Actor_telefone	Este atributo deve ser preenchido com o número de telefone do actor, caso exista.	String		0:1	
Actor_rua	Este atributo deve ser preenchido com o nome da rua e o número de polícia da morada do actor.	String		1:1	
Actor_cidade	Este atributo deve ser preenchido com o nome da cidade da morada do actor.	String		1:1	
Actor_código_postal	Este atributo deve ser preenchido com o código postal da morada do actor.	String		1:1	
Actor_email	Este atributo deve ser preenchido com o endereço de correio electrónico do actor. Assume-se que nenhum actor pode participar num curso sem que possua um endereço de correio electrónico.	String		1:1	
Actor_notificação_tipo	Este atributo deve ser preenchido com a identificação do modo preferencial de comunicação com o actor. Esse modo terá que ser um dos valores permitidos.	Símbolo	SMS, eMail, Telefone, Telefone_móvel, Carta.	1:1	
Actor_telefone_móvel	Este atributo deve ser preenchido com o número de telefone móvel do actor, caso exista.	String		0:1	
Professor_id	Este atributo deve ser preenchido com um identificador que terá que ser único e uma instância da classe Professor.	String		1:1	



## Classe Notificação

**Classe Abstracta Estende:**

:THING

**Instâncias Directas:**

Nenhuma

**Subclasses Directas:**

1. Aviso
2. Alarme1
3. Alarme2
4. Informação

**Documentação da Classe:**

Esta classe abstracta representa os diferentes tipos de notificações que os diversos actores podem receber em certas circunstâncias, durante o funcionamento de um curso.

## Classe Alarme1

**Classe Concreta Estende**

Notificação

**Instâncias Directas:**

Nenhuma

**Subclasses Directas:**

Nenhuma

**Documentação da Classe:**

Esta classe representa a notificação tipo utilizada em situações em que a camada de gestão detecta, antes de atingido o limite temporal para a execução das actividades, que a mesmas ainda não foram executadas pelos destinatários previstos e também não estão em curso.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Notificação_descricao	Este atributo deve ser preenchido com o conteúdo completo da notificação a enviar.	String		1:1	

Notificação_id	Este atributo deve ser preenchido com um identificador que deverá ser único e uma instância da classe Alarme.	String		1:1	
Alarme1_destino	Este atributo deve ser preenchido com identificadores válidos de actores. Estes identificadores devem ser escolhidos de entre o conjunto de todos os identificadores definidos para alunos, professores, elementos das equipas de suporte e responsáveis.	String	Actor_id	0:*	

## Classe Alarme2

### Classe Concreta Estende

Notificação

### Instâncias Directas:

Nenhuma

### Subclasses Directas:

Nenhuma

### Documentação da Classe:

Esta classe representa a notificação tipo utilizada em situações em que a camada de gestão detecta, após atingido o limite temporal para a execução das actividades, que a mesmas não foram executadas pelos destinatários previstos.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Notificação_descricao	Este atributo deve ser preenchido com o conteúdo completo da notificação a enviar.	String		1:1	
Notificação_id	Este atributo deve ser preenchido com um identificador que deverá ser único.	String		1:1	
Alarme2_destino	Este atributo deve ser preenchido com	String	Actor_id	0:*	

	<p>identificadores válidos de actores. Estes identificadores devem ser escolhidos de entre o conjunto de todos os identificadores definidos para alunos, professores, elementos das equipas de suporte e responsáveis.</p>				
--	--	--	--	--	--

## Classe Informação

**Classe Concreta Estende**  
Notificação

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

**Documentação da Classe:**

Este tipo de notificação é considerado pelo facto de poder revelar-se interessante informar alguém relativamente ao sucesso na execução de uma actividade por parte de um determinado actor (principalmente alunos), ou relativamente a qualquer alteração na estrutura do curso.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Notificação_descricao	Este atributo deve ser preenchido com o conteúdo completo da notificação a enviar.	String		1:1	
Notificação_id	Este atributo deve ser preenchido com um identificador que deverá ser único.	String		1:1	
Informação_destino	Este atributo deve ser preenchido com identificadores válidos de actores. Estes identificadores	String	Actor_id	0:*	

	devem ser escolhidos de entre o conjunto de todos os identificadores definidos para alunos, professores, elementos das equipas de suporte e responsáveis.				
--	---	--	--	--	--

## Classe Aviso

**Classe Concreta Estende**  
Notificação

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

### Documentação da Classe:

Esta classe representa a notificação tipo utilizada antes de cada actividade planeada. Desta forma os destinatários de tais actividades serão alertados com antecedência da proximidade de tais actividades.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Notificação_descricao	Este atributo deve ser preenchido com o conteúdo completo da notificação a enviar.	String		1:1	
Notificação_id	Este atributo deve ser preenchido com um identificador que deverá ser único.	String		1:1	
Aviso_destino	Este atributo deve ser preenchido com identificadores válidos de actores. Estes identificadores devem ser escolhidos de entre o conjunto de todos os identificadores	Símbolo	Actor_id	0:*	

	definidos para alunos, professores, elementos das equipas de suporte e responsáveis.				
--	--	--	--	--	--

## Classe Condição

**Classe Concreta Estende**  
:THING

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

### Documentação da Classe:

Esta classe guarda todas as condições que devem ser verificadas pelo LMS no seu processo de lançamento ou libertação de actividades a serem executadas pelos diferentes actores, durante o funcionamento de um curso.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Condição_id	Este atributo deve ser preenchido com um identificador que deverá ser único e uma instância da classe Condição.	String		1:1	
Condição_se_condição	Este atributo deve ser preenchido com a descrição de uma condição a ser verificada pelo LMS, antes que este possa apresentar para execução, a actividade que depende da verificação dessa condição.	String		1:1	

Condição_verificada	Este atributo deve ser preenchido pelo LMS com o valor Falso se a condição descrita no atributo Condição_se_condição não se tiver verificado.	Lógico		1:1	Verda-deiro
---------------------	---	--------	--	-----	-------------

## Classe Grupo

**Classe Concreta Estende**  
:THING

**Instâncias Directas:**  
Nenhuma

**Subclasses Directas:**  
Nenhuma

**Documentação da Classe:**

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Grupo_id	Este atributo deve ser preenchido com um identificador que deverá ser único.	String		1:1	
Grupo_aluno_id	Este atributo deve ser preenchido com identificadores de alunos válidos. Tais identificadores terão que ser parte do conjunto de identificadores que poderemos encontrar na classe Aluno.	String	Aluno	1:*	

## Classe Responsável

**Classe Concreta Estende**  
:THING

**Instâncias Directas:**

Nenhuma

**Subclasses Directas:**

Nenhuma

**Documentação da Classe:**

Esta classe foi incluída pensando fundamentalmente em contextos não académicos de funcionamento de cursos. Nesses contextos (organizacionais) pode considerar-se importante notificar o superior hierárquico (o responsável) de um determinado aluno (funcionário) relativamente ao desempenho do mesmo, em determinadas circunstâncias particulares.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Responsável_id	Este atributo deve ser preenchido com o identificador do responsável, que terá que ser único e uma instância da classe Responsável.	String		1:1	
Responsável_telefone_móvel	Este atributo deve ser preenchido com o número de telefone móvel do responsável, caso exista.	String		0:1	
Responsável_telefone	Este atributo deve ser preenchido com o número de telefone do responsável, caso exista.	String		0:1	
Responsável_notificação_tipo	Este atributo deve ser preenchido com a identificação do modo preferencial de comunicação com o responsável. Esse modo terá que ser um dos valores permitidos.	Símbolo	SMS, eMail, Telefone, Telefone_móvel, Carta.	1:1	
Responsável_email	Este atributo deve ser preenchido com o endereço de correio electrónico do responsável. Assume-se que nenhum responsável	String		0:1	

	pode ser considerado no âmbito de um curso sem que possua um endereço de correio electrónico.				
Responsável_código_postal	Este atributo deve ser preenchido com o código postal da morada do responsável.	String		1:1	
Responsável_cidade	Este atributo deve ser preenchido com o nome da cidade da morada do responsável.	String		1:1	
Responsável_rua	Este atributo deve ser preenchido com o nome da rua e o número de polícia da morada do responsável.	String		1:1	
Responsável_password	Este atributo deve ser preenchido com a "password" do responsável, caso esteja prevista a utilização do sistema por parte do mesmo. Essa "password" deve ser única.	String		0:1	
Responsável_username	Este atributo deve ser preenchido com o "username" do responsável, caso esteja prevista a utilização do sistema por parte do mesmo.	String		0:1	
Responsável_nome	Este atributo deve ser preenchido com o nome do responsável.	String		1:1	

## Classe Supervisor

Classe Concreta Estende  
:THING

Instâncias Directas:  
Nenhuma



**Subclasses Directas:**

Nenhuma

**Documentação da Classe:**

Esta classe representa pessoas que são responsáveis pela execução das diversas actividades. Tais pessoas podem não fazer parte do conjunto dos actores que normalmente concebem as actividades dos cursos. Será contudo a estas pessoas que os diversos actores se deverão dirigir, no sentido do esclarecimento de dúvidas que possam ter quanto à execução das actividades a si destinadas.

Atributo	Documentação	Tipo	Valores/Classes Permitidas	Cardinalidade	Valor por Defeito
Supervisor_id	Este atributo deve ser preenchido com o identificador do supervisor. Esse identificador deverá ser único e uma instância da classe Supervisor.	String		1:1	
Supervisor_nome	Este atributo deve ser preenchido com o nome do supervisor.	String		1:1	
Supervisor_username	Este atributo deve ser preenchido com o "username" do supervisor.	String		1:1	
Supervisor_password	Este atributo deve ser preenchido com a "password" do supervisor. Tal "password" deverá ser única.	String		1:1	
Supervisor_telefone	Este atributo deve ser preenchido com o número de telefone do supervisor, caso exista.	String		0:1	
Supervisor_telefone_móvel	Este atributo deve ser preenchido com o número de telefone móvel do supervisor, caso exista.	String		0:1	

Supervisor_Notificação_tipo	Este atributo deve ser preenchido com a identificação do modo preferencial de comunicação com o supervisor. Esse modo terá que ser um dos valores permitidos.	Símbolo	SMS, eMail, Telefone, Telefone_móvel, Carta.	1:1	
Supervisor_email	Este atributo deve ser preenchido com o endereço de correio electrónico do supervisor. Assume-se que nenhum supervisor pode ser considerado no âmbito de um curso sem que possua um endereço de correio electrónico.	String		1:1	
Supervisor_rua	Este atributo deve ser preenchido com o nome da rua e o número de polícia da morada do supervisor.	String		1:1	
Supervisor_cidade	Este atributo deve ser preenchido com o nome da cidade da morada do supervisor.	String		1:1	
Supervisor_código_postal	Este atributo deve ser preenchido com o código postal da morada do supervisor.	String		1:1	

Tabela 4 – Metadados associados à definição de um LMS.

### 5.5.2 Pontos de ancoragem ao projecto SCORM

Uma das fases importantes deste trabalho consistiu na identificação dos pontos de ancoragem ao projecto SCORM. Com efeito, havia que encontrar pontos da arquitectura proposta pela norma SCORM, que pudessem constituir canais de comunicação ou pontos de integração com a camada de gestão proposta. Foram identificados vários pontos onde essa integração poderia ser realizada, sendo que a mesma implica a alteração de processos já existentes na norma ou a criação de novas operações ainda não consideradas. Também

ao nível do modelo de dados se verificou ser necessárias algumas adaptações, uma vez que aspectos como as diferentes notificações ou os actores “não alunos”, não estão previstos na norma.

Apresentam-se seguidamente algumas das características do modelo SCORM, que dão indicações claras quanto à sua potencial utilização no sentido da integração desejada.

O modelo de dados do RTE contém um conjunto de elementos, relacionados com as interações entre um SCO e o LMS, e que o primeiro pode registar durante a sua execução.

A tabela 5 apresenta os elementos do modelo de dados do RTE.

Data Model Element	Description
Comments From Learner	Contains text from the learner.
Comments From LMS	Contains comments and annotations intended to be made available to the learner.
Completion Status	Indicates whether the learner has completed the SCO.
Completion Threshold	A value against which the measure of the progress the learner has made toward completing the SCO can be compared to determine whether the SCO should be considered completed.
Credit	Indicates whether the learner will be credited for performance in this SCO.
Entry	Contains information that asserts whether the learner has previously accessed the SCO.
Exit	Indicates how or why the learner left the SCO.
Interactions	Defines information pertaining to an interaction for the purpose of measurement or assessment.
Launch Data	Provides data specific to a SCO that the SCO can use for initialization.
Learner Id	Identifies the learner on behalf of whom the SCO instance was launched.
Learner Name	Represents the name of the learner.
Learner Preference	Specifies learner preferences associated with the learner's use of the SCO.
Location	Represents a location in the SCO.
Maximum Time Allowed	The amount of accumulated time the learner is allowed to use a SCO in the learner attempt.
Mode	Identifies the modes in which the SCO may be presented to the learner.
Objectives	Specifies learning or performance objectives associated with a SCO.
Progress Measure	A measure of the progress the learner has made toward completing the SCO.
Scaled Passing Score	The scaled passing score for a SCO.
Score	The learner's score for the SCO.
Session Time	The amount of time that the learner has spent in the current learner session for the SCO.
Success Status	Indicates whether the learner has mastered the SCO.
Suspend Data	Provides information that may be created by a SCO as a result of a learner accessing or interacting with the SCO.
Time Limit Action	Indicates what the SCO should do when the maximum time allowed is exceeded.
Total Time	The sum of all of the learner's learner session times accumulated in the current learner attempt prior to the current learner session.

Tabela 5 – Elementos do modelo de dados do RTE.  
Adaptado de (ADL, 2004c).

A sombreado podem identificar-se os elementos que, de alguma forma, estão mais directamente relacionados com o trabalho realizado no âmbito desta tese, nomeadamente o elemento “*completion\_status*”, utilizado para indicar quando o aluno terminou um SCO.

Recordando o que se apresentou na secção 3.5.3 relativamente ao trabalho realizado no âmbito do projecto ADL SCORM, chega-se de novo ao conceito de árvore de actividades então descrito, e ao que essa estrutura representa em termos da organização dos conteúdos e das actividades de um curso.

A especificação IMS SS, de que a norma SCORM Sequencing and Navigation derivou, descreve a informação de seguimento (*Tracking Information*) que deve ser mantida por cada actividade na árvore de actividades.

Durante uma sessão de aprendizagem, os elementos do modelo de seguimento (*Tracking Model*) vão sendo actualizados por forma a reflectirem as interacções do aluno com o objecto de conteúdo corrente.

Todas as actividades possuem informação de seguimento associada a cada aluno que executou, total ou parcialmente, a actividade.

O modelo de seguimento descreve a informação que deve ser mantida por um LMS relativamente a cada actividade definida para um curso. Essa informação aparece distribuída segundo os seguintes conjuntos:

- **Objective Progress Information:** Descreve o progresso do aluno relativamente ao objecto de conteúdo.
- **Activity Progress Information:** Descreve o progresso do aluno numa dada actividade. Esta informação descreve esse progresso de uma forma cumulativa, ao longo de todos os acessos à actividade, até esta ser dada como concluída.
- **Attempt Progress Information:** Descreve o progresso do aluno numa dada actividade, mas indicando o progresso conseguido em cada acesso a essa actividade.
- **Activity State Information:** Descreve o estado de uma actividade, por árvore de actividade e por aluno.

Enumeram-se de seguida aqueles que pareceram ser os pontos mais indicados para realizar a integração referida, e descreve-se a forma como essa integração deve ser implementada:

- a) Em (ADL, 2004c), documentação relativa ao RTE, é referido que durante a execução de um SCO lançado pelo LMS, aquele detecta uma instância da API e inicia a comunicação entre si próprio e o LMS, através da invocação de determinados métodos da API. Tais métodos encontram-se distribuídos segundo três grandes grupos – Métodos de Sessão, Métodos para Transferência de Dados e Métodos de Suporte.

Os métodos de sessão “*Initialize()*” e “*Terminate()*” são utilizados para dar início e terminar o processo de comunicação, enquanto que os métodos de transferência de dados “*GetValue()*”, “*SetValue()*” e “*Commit()*”, são invocados para gerir os processos de armazenamento e obtenção da informação a utilizar na sessão de comunicação em curso. O método “*SetValue()*” é utilizado para enviar informação do SCO para o LMS, para aí ser guardada.

Alterando as funcionalidades deste método, será possível promover a criação e o registo da informação adequada, no repositório de eventos proposto no âmbito deste trabalho, nos casos em que tal se justifique, isto é, quando uma actividade tiver sido concluída.

- b) Se o SCO, utilizando o elemento *cmi.completion\_status* do Modelo de Dados do RTE, informar que o aluno terminou esse SCO, então a actividade a que o SCO pertencia também deve ser considerada terminada (ADL, 2004c).

Assim sendo, se este mecanismo associado ao SCO for estendido, poderá criar e inserir a informação necessária no repositório de eventos, para que a camada de gestão possa realizar as comparações descritas na secção 5.5 deste documento.

O modelo de dados apresentado na tabela 5 terá também que ser aumentado, por forma a incluir alguns dos elementos inventariados na tabela 4 (metadados de gestão), mormente no que diz respeito ao elemento “Actividade”.

- c) Por outro lado, o elemento do modelo de dados *cmi.time\_limite\_action*, indica como o SCO deve actuar quando o valor desse elemento é excedido.

Sem grande esforço se poderá concluir que o mesmo coincide com os propósitos do elemento de metadados “segundo alarme” que é parte integrante da estrutura proposta neste trabalho e apresentada na figura 31 da secção 5.2.6.

Com efeito, ao ser atingido esse instante, a camada de gestão deve consultar o repositório de “Eventos” no sentido de verificar se existe um registo correspondente à finalização da actividade correspondente.

É com base nessa verificação que a camada de gestão pode concluir da necessidade ou não do envio de uma notificação do tipo “segundo alarme” aos destinatários identificados.

- d) Identificou-se ainda um quarto ponto de integração da camada de gestão proposta neste trabalho, com a norma SCORM. Em (ADL, 2004c), no anexo C, *SCORM Sequencing Behaviour Pseudo Code*, é possível ler-se que deve ser atribuído o valor “verdadeiro” ao atributo *Objective Satisfied Status* sempre que um objectivo é atingido. Trata-se, naturalmente, de uma oportunidade para fazer também a actualização do repositório “Eventos”, já que a execução com sucesso de uma actividade dentro dos limites temporais predefinidos, pode significar que foi satisfeito um objectivo.

## 5.6 Contribuições para o projecto SCORM

Como é sabido, o acrónimo SCORM significa *Sharable Content Object Reference Model*. Daqui se retira imediatamente a ideia base deste projecto de normalização, que como se depreende, está orientado fundamentalmente para os conteúdos, fornecendo orientações quanto à forma de criar pacotes (*content packages*) reutilizáveis<sup>53</sup> e quanto às possibilidades e condições de apresentação e de interacção com esses conteúdos<sup>54</sup>. Apenas com o aparecimento em 2004 do *Sequence and Navigation Book*, passaram a estar

---

<sup>53</sup> Informação disponível no SCORM Content Agregation Metamodel Book (ADL, 2004b).

<sup>54</sup> Informação disponível no SCORM Run-Time Environment Book (ADL, 2004c).

disponíveis indicações relativas às actividades e sua estrutura agregadora – *Activity Tree*, bem como às condições de navegação pelas mesmas.

O resultado do trabalho descrito nesta tese possibilita claramente o enriquecimento das recomendações SCORM, já que passam a estar cobertos aspectos dinâmicos da interacção dos utilizadores com os cursos e os LMS e a gestão em tempo real de tais interacções. Este enriquecimento materializa-se no facto de também aspectos relacionados com as funcionalidades dos LMS passarem a estar cobertas.

Por outro lado, ao contrário do previsto na norma SCORM, propõe-se que não só os alunos sejam vistos como actores relevantes no processo de ensino/aprendizagem mas também os professores e os elementos das equipas de suporte. Facilmente se poderia indicar um conjunto de actividades importantes nesse processo de ensino/aprendizagem, cuja execução é da responsabilidade destes actores e cuja não execução pode igualmente fazer comprometer todo o processo.

Finalmente, propõe-se nesta tese a inclusão do conceito de grupo de trabalho, que não existe na especificação SCORM. Nesta, só o aluno, individualmente, é visto como podendo interagir com os LMS, no sentido da execução das actividades constituintes de um curso.

## 5.7 Validação dos resultados

Um aspecto importante num trabalho deste tipo consiste na validação dos resultados e na retirada de conclusões com base nos mesmos.

No caso presente essa validação não assentou em dados reais, fruto da realização de um trabalho de campo. A razão para a ausência desse tipo de validação prende-se com o facto de não ter sido previsto o desenvolvimento de qualquer protótipo e, portanto, não ter sido possível avaliar os níveis de sucesso dos cursos, fruto do desempenho dos diferentes actores, automaticamente monitorizados por uma camada de gestão construída de acordo com a arquitectura proposta nesta tese.

Mesmo que tal protótipo tivesse sido desenvolvido, a sua validação continuaria a ser difícil de realizar na janela temporal definida para a execução do presente trabalho. Tal dificuldade resultaria do facto de ser necessário encontrar e estudar profundamente um LMS *open source* construído de acordo com as recomendações SCORM existentes, ao qual fosse possível acoplar o novo módulo. Nessa situação, seria necessário executar em

paralelo várias versões dos mesmos cursos, frequentados pelas mesmas populações, com e sem este tipo de monitorização, de forma a ser possível retirar algum tipo de conclusões minimamente aceitáveis.

Ainda assim, dificilmente se conseguiria um conjunto de resultados completamente esclarecedores já que, se não impossível, pelo menos seria muito difícil garantir que os intervenientes tivessem exactamente o mesmo tipo de comportamento em ambas as versões dos cursos. Na melhor das hipóteses teríamos apenas indicadores aproximados, relativos à introdução desta camada de gestão das actividades.

Um outro cenário para a realização da validação das ideias que estiveram na origem do trabalho realizado, consistia na recolha das opiniões de um painel de peritos. Essa abordagem permitiria uma análise qualitativa das opiniões recolhidas e, em consequência dessa análise, a retirada de conclusões acerca do modelo proposto.

Optou-se por uma abordagem deste tipo, levada a cabo através da submissão de um conjunto de artigos derivados do trabalho realizado, a conferências internacionais de áreas diversificadas mas relacionadas de alguma forma com o tema. Foram seleccionadas conferências em que a aceitação dos trabalhos submetidos estivesse dependente da avaliação prévia dos trabalhos por três revisores. As conferências em causa foram as seguintes: ELPUB2006 (10<sup>th</sup> International Conference on Electronic Publishing), EUNIS2006 (12<sup>th</sup> International Conference of European University Information Systems) e ICALT2006 (6<sup>th</sup> IEEE International Conference on Advanced Learning Technologies).

O facto de os referidos artigos terem sido aceites nessas conferências é, por si só, um indicador da validade e oportunidade das ideias que os mesmos comportavam. Esta convicção foi reforçada pelas críticas emitidas pelos revisores e pelo interesse que as apresentações provocaram nas audiências.

Um outro facto que permite concluir que o tema é interessante e oportuno prende-se com o facto de o projecto IMS LD ter passado a incluir no seu nível C (o mais recente), uma componente de notificações em tempo real, que pode sem dúvida evoluir para uma utilização semelhante à que é proposta neste trabalho.

Informalmente, o tema foi também objecto de algumas trocas de impressões com um dos principais elementos do projecto CopperCore (IMS, 2006), o Doutor Daniel Burgos, que sempre manifestou uma opinião positiva quanto ao trabalho que estava a ser realizado



no âmbito desta tese. De referir que o CopperCore é uma ferramenta de empacotamento de conteúdos compatível com a componente CAM da norma SCORM.

# Capítulo 6

## Conclusões

### 6.1 Respondendo às questões de investigação

Terminado o trabalho que levou à escrita da presente tese, impõe-se invocar aquelas que eram as questões de investigação colocadas na secção 1.2.2 e ver até que ponto foi possível dar-lhes resposta e que respostas foi possível dar. Assim, tomando a primeira das questões – Que modelo conceptual deve servir de base à gestão operacional de um ambiente de ensino/aprendizagem – as conclusões a que se chegou são as de que uma camada de gestão de actividades deve possuir a capacidade de, em tempo real, detectar os desvios da execução das actividades programadas, face ao plano dos cursos. A identificação atempada de tais situações tornará possível a emissão de notificações que permitam a execução de medidas correctivas conducentes à recuperação desses desvios, sem necessidade de alterar os planos dos cursos.

Por outro lado, a inclusão de uma funcionalidade capaz de alertar antecipadamente os diversos actores da proximidade do início de uma nova actividade possibilitará também uma melhor gestão dos tempos e dos trabalhos de cada utilizador, por forma a aumentar as possibilidades de sucesso na execução das actividades e, afinal, dos cursos.

Como se viu na secção 5.5 a implementação do modelo deve passar pela inclusão de dois repositórios. Um para caracterização das actividades dos cursos e outro para o registo das interações dos utilizadores com o LMS e com os cursos (os eventos). A camada de gestão propriamente dita, terá que ser capaz de comparar de uma forma continuada o

conteúdo de ambos os repositórios, de forma a detectar quando uma actividade foi ou não executada de acordo com o previsto.

A terceira componente consiste no mecanismo que fará a invocação das funcionalidades de *messaging* do LMS, passando-lhe a informação necessária à emissão das notificações adequadas a cada situação.

Desejavelmente, a identificação de uma situação justificadora de um segundo alarme deveria solicitar a camada de sequenciamento e navegação do LMS, de forma a permitir uma redefinição da sequência de actividades a executar a partir desse momento.

Uma solução como a que foi apontada na secção 5.2.3.6, relativamente à “Execução de actividades de recuperação”, seria uma possibilidade (cf. figura 30).

Convirá neste momento realçar o papel fundamental da estrutura atómica de gestão de actividades em tempo real, tal como foi apresentada na figura 31. Deve lembrar-se que na arquitectura proposta, um curso é uma combinação de unidades daquele tipo, organizadas de forma sequencial e/ou paralela, e que nada do que se descreveu nesta tese poderá ser implementado sem a utilização de tal estrutura básica de gestão de actividades.

Quanto à segunda questão de investigação – Que atributos informacionais devem ser contemplados na gestão operacional de um ambiente de ensino/aprendizagem – pode identificar-se um núcleo base de atributos que têm que fazer parte do modelo. Esses atributos estão relacionados com a estrutura atómica de gestão de actividades apresentada na secção 5.2.6 (cf. figura 31). Sem qualquer dúvida, dado que um curso pode e deve ser visto como uma combinação de elementos daquele tipo, será imprescindível a existência dos atributos associados à estrutura atómica de gestão de actividades, bem como daqueles atributos que caracterizam os diversos actores intervenientes no processo.

A identificação dos atributos necessários é intuitiva e diz respeito aos elementos “Actividade”, “Evento”, “Actor”, “Grupo”, “Alarme” e “Supervisor”, que fazem parte da representação gráfica do esquema XML apresentado na figura 34.

Informação com maior detalhe, relativa a esses atributos, pode ser consultada na tabela 4 desta tese, na secção 5.5.1.

Podendo parecer ser a questão de investigação à qual seria mais difícil dar uma resposta, a terceira questão – Como poderá ser efectuada a integração das funcionalidades de gestão operacional de ensino/aprendizagem na norma SCORM – acabou por permitir a identificação de uma solução relativamente simples.

A leitura do presente documento, nomeadamente da sua secção 5.5, permite concluir que a base para a solução do problema da gestão de actividades de ensino/aprendizagem em tempo real passa pela comparação do plano do curso (repositório Actividades) com o registo das interacções dos actores com o curso (repositório Eventos) e pela identificação de desvios na execução das actividades, face ao planeado.

Como se viu na secção 5.5.2, a norma SCORM prevê a utilização de estruturas de caracterização das actividades de um curso (Activity Tree), bem como o registo das interacções dos utilizadores com essas actividades (Tracking Information).

Ao pensar-se no desenvolvimento de um protótipo que implemente as ideias descritas neste trabalho, apenas será necessário compatibilizar a informação registada pelos métodos da API do RTE, com aquele conjunto de atributos que foi inventariado na tabela 4.

Garantida que esteja a informação necessária, à camada de gestão apenas caberá a realização das comparações já mencionadas e as eventuais solicitações do sistema ou camada de *messaging* do LMS.

## 6.2 Quanto às técnicas de modelação utilizadas

Os capítulos 4 e 5 desta tese estão directamente relacionados com técnicas de representação de conhecimento ou de modelação de sistemas. No capítulo 4 apresentaram-se características genéricas de algumas técnicas bem como a vocação específica que estas apresentam, tendo sido afirmada a vantagem da sua utilização combinada, como forma de captar a maior quantidade de conhecimento possível acerca das diversas vertentes que devem ser analisadas num projecto de arquitectura e desenvolvimento de sistemas de informação.

No capítulo 5 utilizou-se intensivamente um subconjunto de diagramas da linguagem UML, como forma de identificar os requisitos do sistema (diagramas de casos de uso) e de representar os aspectos dinâmicos do sistema (diagramas de sequência), materializados na troca de mensagens entre os elementos dos diagramas.

A UML é hoje aceite universalmente como a linguagem por excelência para suportar a especificação e o desenvolvimento de software, nomeadamente quando se tratar do paradigma orientado a objectos. Embora o desenvolvimento de um protótipo não fosse um objectivo a atingir no âmbito deste trabalho, ao ter-se utilizado os Diagramas de Caso de

Uso e os Diagramas de Sequência, de certa forma garantiu-se a possibilidade de, no futuro, se poder dar continuidade ao projecto, com o objectivo do desenvolvimento da camada de gestão de actividades, integrável num LMS compatível com a norma SCORM.

Ainda no que diz respeito à UML, foi possível confirmar a utilidade dos diagramas de sequência estereotipados, conforme sugerido em (Machado et al, 2005), que substituem os objectos por actores, estereótipos e casos de uso. Tal substituição é possível, aceitável e mesmo desejável, ao nível de abstracção a que este trabalho se limitou, já que não se estava em presença de um projecto completo, em que o objectivo final fosse o desenvolvimento de software. De referir que a um nível de abstracção tão elevado e em fases tão iniciais do processo, é mesmo difícil conseguir identificar que objectos poderão vir a ser necessários na implementação do sistema final.

A representação dos relacionamentos entre as diversas componentes do sistema, as suas solicitações através do envio de mensagens e as reacções resultantes dessas solicitações, ficaram bem documentadas nos diagramas apresentados, tendo ajudado a consolidar a percepção que já se tinha do problema.

A utilização de uma certa forma de diagramas E-R na secção 5.4 (cf. figura 32), importada directamente da ferramenta Microsoft Access, tornou-se uma inevitabilidade, dado que tendo este trabalho sido uma continuação do projecto ODL Toolbox, algumas das ideias recuperadas para esta tese apareceram como consequência lógica do trabalho então realizado. Nessa altura, procurou-se representar o sistema e desenvolver um conjunto de funcionalidades em Microsoft Access, que permitissem fazer uma primeira validação de alguns dos conceitos subjacentes à realização daquele trabalho. Como se referiu anteriormente, esta tese de doutoramento foi uma evolução do trabalho realizado nesse projecto. Daí a utilização do esquema da figura 32 que permite suportar as respostas às questões de competência colocadas e, dados os resultados obtidos, permitiu concluir que o modelo de que se estava a partir, permitia atingir os objectivos pretendidos.

A dada altura da realização do trabalho, e após algumas incursões por diversas abordagens de modelação, as ontologias de domínio pareceram ser uma boa aposta para a captura e representação do conhecimento relativo à implementação da camada de gestão de actividades em ambientes de ensino/aprendizagem baseados na utilização de LMS. A escolha da ferramenta Protégé2000 para a realização das primeiras experiências na área foi mais ou menos natural, já que as referências à mesma, quer na Web, quer na literatura,

eram muitas. Por outro lado, tratava-se de uma ferramenta de custo zero e havia bastante documentação disponível sobre a sua utilização.

A literatura disponível refere a utilização deste tipo de abordagem, aplicado a um vastíssimo leque de problemas nas mais diferentes áreas. Contudo, e porque o próprio problema da inclusão nos LMS de uma camada de gestão operacional não aparece mencionado na bibliografia, não se encontraram referências à utilização de ontologias de domínio aplicadas à referida área. Este facto, conferia à aplicação das ontologias de domínio ao problema em causa, um certo estatuto inovador.

O Protégé2000 para além de se constituir num editor de ontologias com grandes potencialidades e de utilização relativamente fácil, dispõe ainda de uma série de funcionalidades nativas e de outras que resultam da adição de *plugins*, que permitem, por exemplo, a obtenção automática dos elementos de metadados associados à definição de uma ontologia, ou a geração de ficheiros em formato XMI, capazes de permitir a obtenção automática de diversos modelos UML.

Apesar destas características achadas interessantes, chegou-se à conclusão de que os aspectos dinâmicos associados à execução das actividades de ensino/aprendizagem, dificilmente seriam representáveis através de uma ontologia de domínio.

Convirá recuperar neste momento o que se apresentou na secção 4.3 desta tese, dizendo respeito à posição de Noy (Noy, 2001a), segundo quem, a construção de uma ontologia de domínio implica:

- A definição das classes (conceitos) da ontologia;
- Estruturação das classes numa hierarquia taxonómica (subclasses-superclasses);
- Definição para cada classe, dos atributos (*slots*) e descrição dos valores permitidos para esses *slots*;
- Preenchimento dos valores dos *slots* para as instâncias das classes.

Como se pode observar, nenhuma das tarefas associadas à definição de uma ontologia pode ser associada à execução das actividades.

## 6.3 Dificuldades e limitações sentidas

Durante a realização do trabalho descrito nesta tese foram diversas as situações que limitaram ou dificultaram a sua realização. Identificam-se de seguida algumas dessas situações, pela maior importância relativa que tiveram no processo.

Desde o início que ficou claro que o trabalho a realizar seria ainda mais solitário e autónomo do que já é habitual em teses de doutoramento. Esta constatação derivou do facto de quase não existir documentação publicada sobre o tema, para além daquela que é tornada pública por cada um dos projectos de normalização em curso (SCORM, IMS, IEEE, AICC, ARIADNE, etc). Por outro lado, o facto de não ter sido desenvolvido um protótipo que implementasse os conceitos inventariados no decurso do trabalho, levou a que se mantivesse até hoje a dúvida sobre a eficácia da proposta constante deste trabalho.

Um outro tipo de dificuldade sentida, prendeu-se com a identificação da melhor abordagem para especificar a camada de gestão. Foram feitas diversas incursões por técnicas e metodologias que, numa primeira fase, pareciam ser a solução para o problema mas que, mais tarde se vinham a revelar-se inúteis ou de muito pouca utilidade no contexto do trabalho a desenvolver. É o caso das ontologias de domínio, matéria que durante bastante tempo foi vista como podendo ser o elemento aglutinador de todo o trabalho de especificação da camada de gestão, mas que a dada altura acabou por se concluir ser insuficiente, dados os objectivos do trabalho.

Constituiu ainda uma dificuldade, o facto de a UML ser uma linguagem que utiliza regras descritivas em vez de regras prescritivas (Fowler, 2004). Isto faz com que, na prática, a sua utilização seja menos formal do que pretende ser, já que quando se tenta consolidar a aprendizagem pessoal com os exemplos disponíveis na literatura, não se encontra uma utilização uniformizada das várias técnicas e modelos propostos.

## 6.4 Trabalho futuro

Não é possível quantificar a verdadeira dimensão das vantagens introduzidas pelo presente trabalho nos processos de ensino/aprendizagem suportados por LMS baseados na Web, sem que se possam implementar e utilizar os conceitos e mecanismos de gestão identificados e propostos nesta tese. É portanto inevitável, o desenvolvimento num futuro

mais ou menos próximo, de um protótipo passível de ser fisicamente integrado numa plataforma compatível com as normas SCORM.

Para além desta integração, será fundamental a utilização de um tal LMS em situações o mais parecidas possível com outras pertencentes ao passado e bem conhecidas quanto ao seu funcionamento e resultados. Só desta forma se poderão retirar conclusões sólidas quanto à correcção dos pressupostos de que se partiu para a realização do presente trabalho, bem como da adequação da solução proposta ao problema colocado originalmente.

É importante neste momento fazer uma referência ao facto de todo o trabalho realizado ter sido orientado no sentido de uma eventual integração com plataformas de ensino/aprendizagem compatíveis com a norma SCORM. Porém, revisitando a secção 3.2.3 desta tese, no seu ponto “Learning Design Specification” é possível ler que o projecto IMS Learning Design se podia ter constituído numa alternativa como referencial para a realização desta tese.

Poderá no futuro constituir um desafio interessante, tentar-se a integração da camada de gestão proposta neste trabalho, em plataformas compatíveis com a referida especificação.

Um projecto desse tipo teria até algumas vantagens quando comparado com a hipotética integração com um LMS compatível com a norma SCORM. É que, no âmbito do projecto IMS LD, já existem estruturas e dispositivos, que aproximam mais essa especificação, daqueles que eram os pressupostos da realização desta tese. É o caso, por exemplo, da componente “Notificação”, já contemplada no nível C da especificação.

Embora uma tal integração seja possível, seriam necessárias algumas adaptações ao que é proposto no projecto IMS LD, já que existe uma diferença significativa entre os objectivos das notificações previstas nesse projecto, e o que foi proposto neste trabalho.

No IMS LD uma notificação pode ter como destinatário, um actor ou um componente do sistema, enquanto que no presente trabalho apenas se identificaram como destinatários de notificações, os diversos actores que participam num curso via LMS.

Por outro lado, as notificações que foram propostas foram identificadas como um processo necessário para permitir corrigir percursos ou situações anormais (desvios face ao previsto), durante a execução das actividades de um curso, e decorrentes do incumprimento do plano predefinido. Como se viu, tais notificações, a justificar-se, implicam a solicitação



da camada de *messaging* do sistema, com a passagem dos parâmetros identificadores da mensagem a enviar, do(s) destinatário(s) da mesma e do modo preferencial de comunicação com os destinatários.

Também no que diz respeito ao nível C do projecto IMS LD, existe a possibilidade de o *learning designer* especificar o envio de mensagens e de atribuir novas actividades, em função da ocorrência de certos eventos, sendo que o único tipo de notificação previsto é a mensagem por correio electrónico.

Não parece contudo, que uma compatibilização entre as duas abordagens pudesse ser de implementação muito complicada.

Outros aspectos a ter em conta num eventual projecto de integração da camada de gestão operacional de actividades ensino/aprendizagem numa plataforma compatível com a especificação IMS LD podem ser considerados.

Aparentemente o modelo proposto neste trabalho apresenta uma maior rigidez, quando comparado com o que é proposto no projecto IMS LD, já que o mesmo assenta num processo de execução de actividades com inícios e fins previamente definidos.

Em todo o caso, se no modelo proposto fossem introduzidas as características disponibilizadas pelo IMS LD, que permitem alterar dinamicamente a configuração de um curso, obter-se-ia ainda assim, uma estrutura composta de um conjunto de actividades que pode perfeitamente ser enquadrada pelo modelo que se propõe no presente trabalho. Isto é, seja qual for o encaixe temporal de uma actividade, esta pode ser sempre precedida de um aviso e, caso não seja executada dentro da janela temporal predefinida, provocará o envio de um alarme. Apenas um formato de curso segundo o qual a execução das diversas actividades fosse puramente aleatória, criaria algumas dificuldades no funcionamento do modelo que aqui se propõe. Mas mesmo nesse caso, haveria uma solução. O conjunto de todas essas actividades seria uma “macro actividade”, esta sim, com possibilidade de aviso prévio e de alarmes, quer antes de atingido o fim do curso (primeiro alarme), quer na situação em que todo o conjunto das actividades não fosse executado dentro do tempo previsto (segundo alarme).

# Referências

Abecker, A., van Elst, L. (2004). Ontologies for Knowledge Management. in Handbook on Ontologies, Springer-Verlag, S. Staab & R. Studer (eds), pp.435-454.

ADL - Advanced Destributed Learning (2004a). SCORM 2004 Overview. <http://www.adlnet.gov/scorm/20043ED/Documentation.cfm>. (Último acesso em 15/05/2006).

ADL - Advanced Destributed Learning (2004b). SCORM Content Aggregation Model. <http://www.adlnet.gov/scorm/20043ED/Documentation.cfm>. (Último acesso em 15/05/2006).

ADL - Advanced Destributed Learning (2004c). SCORM Run-Time Environment Version 1.3. <http://www.adlnet.gov/scorm/20043ED/Documentation.cfm>. (Último acesso em 15/05/2006).

ADL - Advanced Destributed Learning (2004d). SCORM Sequencing and Navigation Version 1.3. <http://www.adlnet.gov/scorm/20043ED/Documentation.cfm>. (Último acesso em 15/05/2006).

AICC (2006a). Sítio institucional. <http://www.aicc.org>. (Último acesso em 15/08/2006).

AICC (2006b). AICC Subcommittees. <http://www.aicc.org/pages/aicc2.htm#CMI>. (Último acesso em 15/07/2006).

AICC – Aviation Industry Computer Based Training Committee (2004). CMI Guidelines for Interoperability. <http://www.aicc.org/pages/aicc3.htm#PUB1>. (Último acesso em 26/11/2006).

AIT, (2004). eXe – eLearning XHTML editor. AIT - Auckland University of Technology. <http://exelearning.org/>. (Último acesso em 17/11/2006).

Alani, H. (2003). TGVizTab: An Ontology Visualization Extension for Protégé in Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, USA.

Almeida, M.B. & Bax, M.P. (2003). Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Ci. Inf.*, Brasília, v. 32, nº 3, pp. 7-20.

Amorim, R.M.L., Sánchez, E. (2006). Semantic Modeling of the IMS LD Level B Specification. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Annamalai, M., Sterling, L. (2003). Guidelines for Constructing Reusable Domain Ontologies. Department of Computer Science & Software Engineering, The University of Melbourne, Victoria, Australia.

<http://oas.otago.ac.nz/OAS2003/papers/oas03-annamalai2.pdf>.

Angele, J.G.L. (2004). "Ontologies in F-logic" in Handbook on Ontologies. Berlin, Springer-Verlag, 2004, pág. 29:50.

Ariadne Foundation for the European Knowledge Pool (2006). <http://www.ariadne-eu.org>. (Último acesso em 13/08/2006).

Batista, A.A. (2002). Informattica Online – Um Enquadramento para a Publicação em Linha de Revistas Científicas Electrónicas. Tese de doutoramento submetida à Universidade do Minho.

Barlas, C. (2006). Digital Rights Expression Languages. JISC Technology and Standards Watch. [http://www.jisc.ac.uk/uploaded\\_documents/TSW0603.pdf](http://www.jisc.ac.uk/uploaded_documents/TSW0603.pdf). (Último acesso em 12/08/2006).

Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A., Tielemans, G. (2006). Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in MOODLE LMS. <http://www.jime.open.ac.uk/2005/02/berggree-2005-02.pdf>. (Último acesso em 19/07/2006).

Booch, G., Rumbaugh, J., Jacobson, I. (1999a). The Unified Modeling Language User Guide. Addison Wesley Longman, Inc., Massachusetts, USA.

Booch, G., Rumbaugh, J., Jacobson, I. (1999b). The Unified Modeling Language Reference Manual. Addison Wesley Longman, Inc., Massachusetts, USA.

Borst, W (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Brase, J.W.N. (2004). Ontologies and Metadata for eLearning. in Handbook on Ontologies, Springer-Verlag, S. Staab & R. Studer (eds), pp.555-573.

Buendía, F.A.H. (2006). An evaluation framework for e-learning platforms based on educational standard specification. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

- Burgos, D.M.S. (2006). Adaptive e-Learning Methods and IMS Learning Design: An Integrated Approach. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Caeiro-Rodríguez, M., Llamas-Nistal, M. (2006). A Proposal of Separation of Concerns in EMLs and its Relation With Learning Design. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Corcho, O., Gómez-Pérez, A. (2000). A Roadmap to Ontology Specification Languages. In Dieng R, Corby O. (eds). 12<sup>th</sup> International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. Springer-Verlag, Berlin, Germany, pp 138-153.
- Costa, M.A.C. (2002). A Inteligência Artificial e o Direito –Uma Abordagem Baseada em Ontologias. Tese de Doutorado Submetida à Universidade do Minho.
- Costagliola, G.F.F., Fuccella, V. (2006). Case Studies on the Support of Computer Managed Instruction Functionalities in e-Learning Systems. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Cristea, A.D.B. (2006). Authoring Adaptive Hypermedia and IMS Learning Design: A Possible Understanding?. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Crubézy, M. (2002). The Protégé Axiom Language and Toolset ("PAL"). <http://protege.stanford.edu/plugins/paltabs/pal-documentation/>. (Último acesso em 13/07/2004).
- Dehors, S., Kuntz, R.D. (2006). Reusing Learning Resources Based on Semantic Web Technologies. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- deMarco, T. (1978). Structured Analysis and System Specification. Englewood Cliffs, NJ: Prentice Hall.
- Doan, B.Y.B. (2006). An Educational System Based on Several Ontologies. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Downes, S. (2000). Learning Objects: Resources For Distance Education Worldwide. In International Review of Research in Open and Distance Learning, Volume 2, Number 1.
- Eriksson, H., Penker, M. (1998). UML Toolkit. John Wiley & Sons, Inc, New York, USA.
- Essalmi, F.L. (2006). Graphical UML View from Extended Backus-Naur Form Grammars. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Fowler, M. (2004). UML Distilled Third Edition - A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, Pearson Education, Inc., Boston, USA.

Gascueña, J.M.; González, P. (2006). Domain Ontology for Personalized e-Learning in Educational Systems. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Goh, A., Tou, J.X. (2005). Computer Managed Instruction Interoperability in E-Learning Systems. Proceedings of Web-based Education~WBE 2005~, Grindelwald, Switzerland 2/21/2005.

Gómez-Pérez, A. ; Fernández-Lopez, M.; Corcho, O. (2004). Ontological Engineering. Springer-Verlag, London.

Gounon, P.P.L., Dubourg, X. (2006). An IMS LD Editor to Describe a Tutoring Activity in an On-line Training. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, vol.5: 199-220.

Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. PhD Thesis, University of Twente, The Netherlands. Published as the book "Ontological Foundations for Structural Conceptual Models", Telematica Instituut Fundamental Research Series.

Guizzardi, G., Falbo, R.A., Filho, J.G.P. (2001). Using Objects and Patterns to Implement Domain Ontologies. Computer Science Department, Federal University of Espírito Santo, Vitória – ES – Brazil. Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands.  
<http://wwwhome.cs.utwente.nl/~guizzard/SBES2001vf.pdf>.  
(Último acesso em 19/05/2003).

Haynes, J.A. (2005). Sequencing and Navigation in Simulation-based Training. <http://www.adlnet.gov/downloads/32.cfm>. (Último acesso em 13/02/2006).

Horridge M., Knublauch, H., Rector, A., Stevens, R., Wroe, C. (2004). A Practical Guide to Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0. The University of Manchester and Stanford University.  
<http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>  
(Último Acesso em 23/02/2005).

Hughes, B.R.A.F. (2006). Metadata Challenges for Situational Properties of Learning Objects. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

IEEE LTSC (2006). Institutional Information Site. <http://ieeeltsc.org>.  
(Último acesso em 20/07/2006).

IEEE LTSC (2004). IEEE 1484.11.1 Draft Standard for Learning Technology – Data Model for Content Object Communication. <http://ieeeltsc.org>.  
(Último acesso em 10/04/2006).

IEEE LTSC (2002). IEEE 1484.12.1 Draft Standard for Learning Object Metadata. [http://ltsc.ieee.org/wg12/files/LOM\\_1484\\_12\\_1\\_v1\\_Final\\_Draft.pdf](http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf).  
(Último acesso em 13/07/2005).

IMS Global Learning Consortium Inc (2001a). IMS Learning Resource Meta-data Best Practice and Implementation Guide.  
[http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd\\_bestv1p2p1.html](http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_bestv1p2p1.html).  
(Último acesso em 19/12/2005).

IMS Global Learning Consortium Inc (2001b). About IMS - Background.  
<http://www.imsglobal.org/background.html>.  
(Último acesso em 23/05/2006).

IMS Global Learning Consortium Inc (2003a). IMS Learning Design Information Model.  
[http://www.imsglobal.org/learningdesign/ldv1p0/imslid\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html).  
(Último acesso em 19/12/2005).

IMS Global Learning Consortium Inc (2003b). IMS Simple Sequencing Best Practice and Implementation Guide.  
[http://www.imsglobal.org/simplesequencing/ssv1p0/imsss\\_bestv1p0.html](http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_bestv1p0.html).  
(Último acesso em 19/12/2005).

IMS Global Learning Consortium Inc (2003c). IMS Simple Sequencing Information and Behavior Model.  
[http://www.imsglobal.org/simplesequencing/ssv1p0/imsss\\_infov1p0.html](http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_infov1p0.html).  
(Último acesso em 19/12/2005).

IMS Global Learning Consortium Inc (2003d). IMS Simple Sequencing XML Binding.  
[http://www.imsglobal.org/simplesequencing/ssv1p0/imsss\\_bindv1p0.html](http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_bindv1p0.html).  
(Último acesso em 19/12/2005).

IMS Global Learning Consortium Inc (2006). CopperCore IMS Learning Design Engine Ver.3 (2006). <http://coppercore.sourceforge.net>. (Último acesso em 11/07/2006).

Janezko, D. (2006). Dialogue-Based Authoring of Units of Learning. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Jovanovic, J.C.K., Gasevic, D., Richards, G. (2006). Learning Object Context on the Semantic Web. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Kim, H., Hibbert, D., Kinshuk, (2006). Developing a Learning Management System on the IMS Learning Design Specification. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Klebl, M. (2004). IMS Learning Design: First-hand Experience in Creating Courses. [http://dspace.ou.nl/bitstream/1820/235/2/MKL\\_firsthand\\_imsld\\_2004.pdf](http://dspace.ou.nl/bitstream/1820/235/2/MKL_firsthand_imsld_2004.pdf). (Último acesso em 13/03/2006).

Kleshchev, A.S., Artemjeva, I.L. (2001). A Structure of Domain Ontologies and their Mathematical Models. Expert System Department, Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of the Sciences, Vladivostok, Russia. <http://www.iacp.dvo.ru/es/publ/154.pdf> (Último acesso em 18/10/2004).

Knublauch, H. (2004). Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario With Protégé/OWL. Stanford Medical Informatics, Stanford University, CA. <http://smi-web.stanford.edu/people/holger/publications/MDSW2004.pdf>. (Último acesso em 16/06/2005)

Kogut, P., Cranefield, S., Hart, L., Dutra, M., Baclawski, K., Kokar, M., Smith, J. (2002). UML for Ontology Development. Lockheed Martin Management & Data Systems. University of Ontago, GRC International, Sandpiper Software, Inc, Northeastern University, Mercury Computer. <http://ubot.lockheedmartin.com/ubot/papers/publication/KER4.doc>. (Último acesso em 12/05/2004).

Koper, R. (2005). Current Research in Learning Design. <http://dspace.ou.nl/bitstream/1820/456/1/ETS-final-editorial.pdf>. (Último acesso em 22/10/2005).

Kraan, W. (2004). CopperCore to power Learning Design implementations. CETIS staff. <http://zope.cetis.ac.uk/content2/20040126154220/printArticle>. (Último acesso em 20/09/2006).

Le Pallec, X., Marvie, R., Nebut, M., Tarby, J.C. (2006). Supporting Generic Methodologies to Assist IMS LD Modeling. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Lee, M., Wang, T.I. (2006). An Ontological Approach for Semantic-Aware Learning Object Retrieval. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Lundgren-Cayrol, K.O.M., Paquette, G., Léonard, M., de la Teja, I. (2006). Implementation and Deployment Process of IMS Learning Design: Findings from the Canadian IDLD Research Project. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

- Machado, R.J., Lassen, K.B., Oliveira, S., Couto, M., Pinto, P. (2005). Execution of UML Models with CPN Tools for Workflow Requirements Validation. Proceedings of the 6th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools - CPN'05, pp. 231-250, Aarhus, Denmark.
- Maedche, A., Staab, S. (2002). *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, Boston, Massachusetts.
- Manso, V.R.R., Palau, C. (2006). Preparing Proprietary Systems for Continuous Education e-Learning for Interoperability: Exporting to SCORM. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Mellor, S. (2005). *Agility in Modeling*. Mentor Graphics Corporation. <http://www.omg.org/docs/omg/05-04-02.pdf>. (Último acesso em 14/04/2006).
- Meyer, M.C.R., Steinmetz, R. (2006). Supporting Modularization and Aggregation of Learning Resources in a SCORM Compliance Mode. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Mizoguchi, R.; Vanwelkenhuysen, J.; Ikeda, M. (1995). Task Ontology for Reuse of Problem Solving Knowledge. in *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing* (ed. N. Mars), pp. 46-57, IOS Press, Amstardam, Neederlands.
- Mizoguchi, R. (2004). Ontology Engineering Environments. In *Handbook on Ontologies*, Springer-Verlag, S. Staab & R. Studer (eds), pp.275-295.
- Moodle (2005). Moodle Docs. [http://docs.moodle.org/en/About\\_Moodle](http://docs.moodle.org/en/About_Moodle). (Último acesso em 15/10/2003).
- Motelet, O.N.B. (2006). Hybrid System for Generating Learning Object Metadata. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Noy, N.F., McGuinness, D.L. (2001a). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA. [http://protege.stanford.edu/publications/ontology\\_development/ontology101.html](http://protege.stanford.edu/publications/ontology_development/ontology101.html). (Último acesso em 12/01/2005).
- Nadolski, R., van der Vegt, W., Koper, R. (2006). Conformance Testing, the Elixer Within the Chain for Learning Scenarios and Objects. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Noy, N.F., Ferguson, R.W., Musen, M. A. (2001b). The knowledge model of Protege-2000: combining interoperability and flexibility. Stanford Medical Informatics, Stanford University, Stanford, CA. [http://www-smi.stanford.edu/pubs/SMI\\_Reports/SMI-2000-0830.pdf](http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-2000-0830.pdf) (Último acesso em 15/10/2003).



- Nunes, M., & O'Neill, H. (2001). *Fundamental de UML*. FCA Editora de Informática, Lisboa, Portugal.
- Pinto, C.S. (1998). *Ambientes de Trabalho Cooperativo no Ensino a Distância*. Dissertação de Mestrado submetida à Universidade do Minho.
- Pinto, C.S., Conde, A., Ramos, F. (2004). *Definindo a Camada de Gestão Operacional de Cursos das Plataformas de e-Learning, com Base Numa Ontologia de Domínio*. Proceedings da conferência eLES'04 (eLearning no Ensino Superior 2004).
- Pinto, C.S., Ramos, F. (2006). *Enhancing Web Supported Learning by Adding a Management Layer to SCORM Compliant LMS*. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).
- Pinto, C.S., Ramos, F. (2006). *Enhancing Web Supported Learning in Higher Education by Adding a Management Layer to LMS*. Proceedings of the 12th International Conference of European University Information Systems (EUNIS2006).
- Pinto, C.S., Ramos, F. (2006). *Modelling a Layer for Real-Time Management of Interactions in Web-Based Distance Learning*. Proceedings of the 10th International Conference on Electronic Publishing (ELPUB2006).
- Qin, J., Hernández, N. (2004). *Ontological Representation of Learning Objects: Building Interoperable Vocabulary and Structures*.  
[http://web.syr.edu/~jqin/pubs/LOV\\_JASIST2006.pdf](http://web.syr.edu/~jqin/pubs/LOV_JASIST2006.pdf). (Último Acesso em 10/01/2005).
- Ramalho, J.C., Librelotto, G.R., Henriques, P.R. (2003). *Ontology driven Websites Metamorphosis: a framework to specify and manage ontology driven websites*. Proceedings of the 7th ICC/IFIP International Conference on Electronic Publishing (ELPUB 2003). Universidade do Minho, Guimarães, Portugal.
- Ramos, F., Conde, A., Neves, L., Moreira, A. (2001). *Towards a Reference Framework for eLearning Management*. Distance Learning Administration Conference (DLA2001), Callaway Gardens, Georgia, USA.
- Ramos, F., Moreira, A., Santos, A., Conde, A., Neves, L., Peixinho, F., Pinto, C.S. (2002). *Tracing dynamic behaviours in distributed learning communities*. Proceedings of the 16th Australian International Education Conference.
- Ramos, F., Moreira, A., Santos, A., Conde, A., Neves, L., Peixinho, F., Pinto, C.S. (2003). *ODL Toolbox: Ferramentas para Apoio à Gestão de Sistemas de Ensino a Distância*. Ref<sup>o</sup> POCTI/COM/33057/99. Fundação Ciência e Tecnologia (FCT).
- Rehak, D.R., Dodds, P., Lannom, L. (2005). *A Model and Infrastructure for Federated Learning Content Repositories*.  
<http://cordra.net/information/publications/2005/www2005/cordrawww2005.pdf>.  
(Último Acesso em 15/03/2006).

Reload - Reusable eLearning Object Authoring and Delivery (2004). Project Summary. <http://www.reload.ac.uk/summary.html>. (Último acesso em 28/10/2006).

Santos, M.Y.C.A. (2001). PADRÃO - Um Sistema de Descoberta de Conhecimento em Bases de Dados Geo-referenciadas. Tese de doutoramento submetida à Universidade do Minho.

Santos, O.A., Ramos, F. (2002). “eLearning on demand: alguns problemas e soluções” in Internet e Educação a Distância. Universidade Federal da Bahia, EDUFBA, 2002, pág. 93:117.

Scholten, H., Beulens, A.J.M. (2002). Ontologies to Structure Models and Modeling Tasks. 16<sup>th</sup> JISR-IIASA Workshop on Methodologies and Tools for Complex System Modeling and Integrated Policy Assessment, IIASA, Laxemburg, Austria. <http://www.info.wau.nl/research%20projects/pub-pdf/hs-ab-CSM02.pdf>.

Shankar, R.D., Musen, M.A. (2002). Use of Protégé-2000 to Encode Clinical Guidelines. Stanford Medical Informatics, Stanford School of Medicine, Stanford, CA. <http://anslese.stanford.edu:8080/smi-web/reports/SMI-2002-0944.pdf> (Último acesso em 15/10/2003).

Silva, A.M.R. (1999). Agentes de Software na Internet - A Próxima Geração de Aplicações para a Internet. Edições Centro Atlântico, Portugal.

Silva, A., Videira, C. (2001). UML - Metodologias e Ferramentas CASE. Edições Centro Atlântico, Lisboa, Portugal.

Silva, A., Figueiredo, A.D. (2002). Teaching Management to Engineering Students: Acting as a Learning Organization. CISUC – Centre of Informatics and Systems of the University of Coimbra, Coimbra, Portugal.

Staab, S., Studer, R. (2004). Handbook on Ontologies. International Handbooks on Information Systems. Springer.

Studer, R., Benjamins, V., Fensel, D. (1998). Knowledge Engineering, Principles and Methods in Data & Knowledge Engineering, vol. 25, n°1-2, pp. 161-197, Elsevier Science, Amsterdam.

Tattersall, C.E.A. (2005). How to use IMS Learning Design and SCORM 2004 together. Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006).

Uschold, M.M.G. (1996). ONTOLOGIES: Principles, Methods and Applications. Knowledge Engineering Review Volume 11 (Number 2, June 1996).

van Duzer, J.K.D.M. (2005). Blackboard vs. Moodle - A Comparison of Satisfaction with Online Teaching and Learning Tools. [http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CSU\\_HSUS/H050215M.pdf](http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CSU_HSUS/H050215M.pdf). (Último Acesso em 10/04/2006).

van Heijst, G., Schreiber, A., Wielinga, B. (1997). Using Explicit Ontologies in KBS Development. *International Journal of Human-Computer Studies*, vol.46, n° 2/3, pp. 183-192.

Vogten, H.H.M., Nadolski, R., Tattersall, C. (2006). CopperCore Service Integration - Integrating IMS Learning Design and IMS Question and Test Interoperability. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006)*.

Vossen, G.P.W. (2006). Towards the Next Generation of e-Learning Standards: SCORM for Service-Oriented Environments. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006)*.

W3C (2004). OWL Web Ontology Language Overview – W3C Recommendation. <http://www.w3c.org/TR/owl-features/>. (Último acesso em 15/10/2005).

Wu, T., Chang, M., Heh, J. (2006). Developing a Learning Notification Express Delivery Service for Students Based on Petri-Net. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006)*.

Yang, S.J.H., Chen, I.Y.L. (2006). Providing Context Aware Learning Services to Learners with Portable Devices. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006)*.