

# Definindo a Camada de Gestão Operacional de Cursos das Plataformas de e-Learning, com Base Numa Ontologia de Domínio

Pinto C.A.S.<sup>1</sup>, Conde A.<sup>2</sup>, Ramos F.M.S.<sup>3</sup>

<sup>1</sup> Universidade do Minho, Guimarães, Portugal

<sup>2</sup> IdeiaPrima, Braga, Portugal

<sup>3</sup> Universidade de Aveiro, Aveiro, Portugal

**Resumo.** O presente artigo pretende descrever o processo que tem sido seguido no sentido da obtenção de um referencial para a camada de gestão on-line dos processos de e-learning. Tal camada deverá ser desenvolvida por forma a poder integrar-se no todo que constitui uma qualquer plataforma de ensino/aprendizagem utilizando tecnologias de informação e comunicação, nomeadamente aquelas que assentam o seu funcionamento nos serviços da Internet. Esta integração será viabilizada a partir da utilização de de uma abordagem comum àquela que já se verifica, relativamente à especificação das restantes camadas, em particular no âmbito do projecto SCORM.

Existem diversas abordagens capazes de permitir a obtenção destes objectivos, isto é, de permitirem a obtenção dos metadados necessários à completa caracterização da camada de gestão operacional de cursos em plataformas de e-Learning. Para levar a cabo este trabalho, recorreremos à escrita de uma ontologia de domínio, aplicável a este caso concreto. Tal abordagem, para além de conjugar em si todas as características de formalismo e rigor desejáveis, permite ainda um entendimento muito fácil do modelo desenvolvido. Para além disso, a sua tradução para XML pode ser também obtida com um mínimo de esforço, possibilitando a divulgação do documento num formato de utilização universal na Web.

## 1. Introdução

A generalidade das plataformas de e-Learning apresenta, de uma forma mais ou menos clara, um conjunto de camadas destinadas a suprir as necessidades que se verificam aos diversos níveis de utilização das mesmas. Por exemplo, ao nível dos conteúdos, ao nível dos interfaces, etc. Diversas organizações e consórcios se têm preocupado nos últimos anos com a definição formal de padrões que permitam reger o desenvolvimento futuro de plataformas de e-Learning, tendo no horizonte preocupações de reuso de componentes e interoperabilidade de sistemas. Nesse contexto, diversos projectos estão em curso. É o caso do IMS (Instructional Management System) consortium, do IEEE LTSC/LOM (Learning Technology Standards Committee - Learning Object Metadata), do projecto europeu Ariadne, do projecto ADL/SCORM (Advanced Distributed Learning - Sharable Content Object Reference Model), etc. Contudo, aspectos relacionados com a gestão operacional dos cursos em ambientes de e-Learning têm ficado de fora desses esforços. Acredita-se que a não execução de determinadas tarefas dentro das janelas temporais definidas pelos planos dos cursos possa comprometer a execução de outras tarefas ou o trabalho de outros elementos participantes, constituindo este aspecto um factor crítico de sucesso na operacionalização de um curso. É, pois, de importância vital para o sucesso dos mesmos,

que existam mecanismos de controlo automático das interacções dos diversos actores com as plataformas de e-Learning.

Embora existam algumas plataformas que, de algum modo, já apresentem algumas dessas funcionalidades, tais plataformas implementam-nas de um modo proprietário, inviabilizando o acima referido reuso e interoperabilidade.

O projecto SCORM surgiu como forma de aproveitar, sistematizar e estender o trabalho que vinha sendo levado a cabo no âmbito dos outros projectos/organizações e com o objectivo principal de se constituir no modelo de referência a nível internacional para o desenvolvimento de plataformas e de conteúdos de e-Learning.

## 2. Justificação Para a Realização do Trabalho

Tentando aproveitar todas as características de formalismo, rigor e universalidade de entendimento colocadas à disposição pela utilização de metadados, e a facilidade de tradução destes para uma linguagem considerada standard na Web, o XML, a ADL arquitectou através da figura de um conjunto de livros, cada um descrevendo uma das camadas, toda a estrutura envolvida nas plataformas de e-Learning.

A Figura 1 mostra a arquitectura baseada nos referidos livros, realçando em cada caso a origem dos trabalhos de normalização levados a cabo.

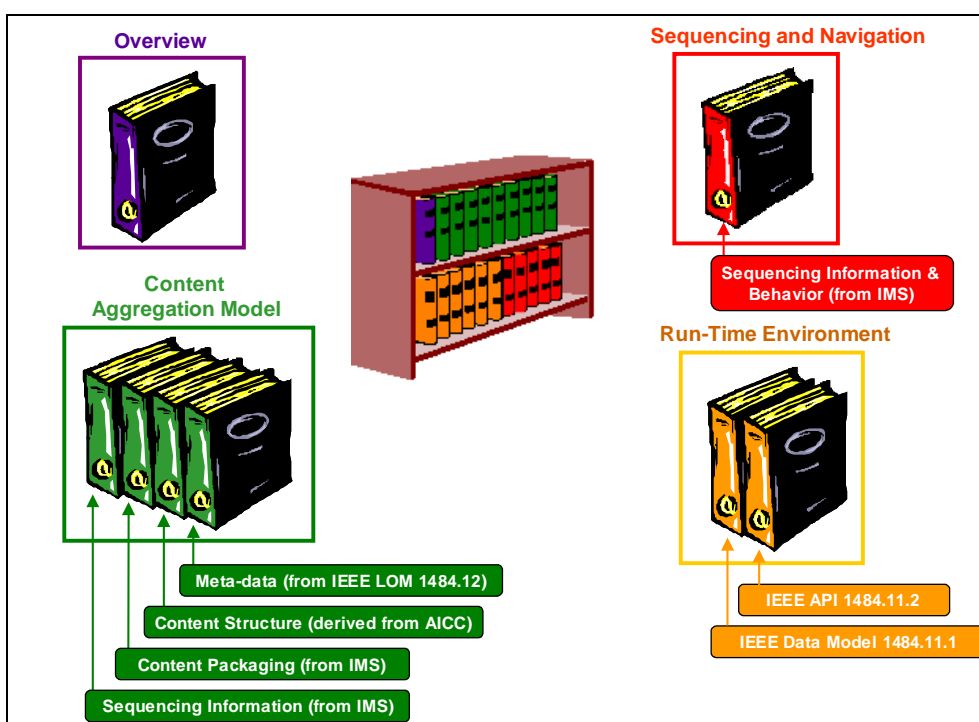


Figura 1 – Organização das normas SCORM como uma colecção de livros de especificações (Adaptado de SCORM, 2004)

Contudo, 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 interações dos diversos actores com as plataformas de e-Learning, durante a operacionalização dos cursos.

Foi desta forma que surgiu a ideia de propor a inclusão de um novo livro, relativo à especificação da camada relativa à gestão operacional dos cursos.

### 3. O trabalho já realizado

O trabalho que nos propusemos tem vindo a ser desenvolvido desde há dois anos no âmbito do projecto ODL Toolbox: Ferramentas para Apoio à Gestão de Sistemas de Ensino a Distância [2] e começou com a identificação das diferentes fases na arquitectura e operacionalização de um curso à distância, com realce para a caracterização das diversas actividades normalmente existentes nesse contexto, observáveis e passíveis de serem controladas. Para cada uma dessas actividades foram identificados os actores envolvidos, as pré-condições para a sua execução, os procedimentos associados e os avisos e alarmes a gerar quando as situações o justificam, bem como os destinatários de tais notificações.

Durante esta fase de realização do trabalho identificámos a unidade elementar de qualquer curso, em termos das actividades, das entidades envolvidas, das acções a desencadear e dos mecanismos de controle necessários.

A Figura 2 representa a unidade elementar do que consideramos ser a estrutura global de um curso, em termos operacionais. Ou seja, um curso é composto por actividades calendarizadas, a executar por um ou mais dos actores envolvidos no curso, e devem ocorrer dentro de janelas temporais previstas. Antes de iniciada qualquer actividade, o sistema deve emitir um aviso e, em caso de incumprimento da actividade, os intervenientes relevantes devem ser notificados como resultado do despoletar de um alarme.

Os procedimentos a adoptar perante uma situação de alarme serão, obviamente, específicos de cada situação concreta.

Da mesma forma, os espaços temporais que separam um aviso, do início de uma actividade e o fim de uma actividade, da emissão de um alarme, serão também variáveis e a definir em cada situação.

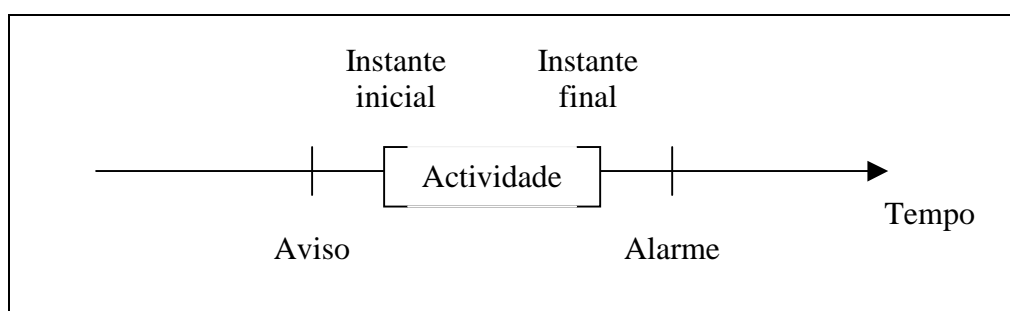


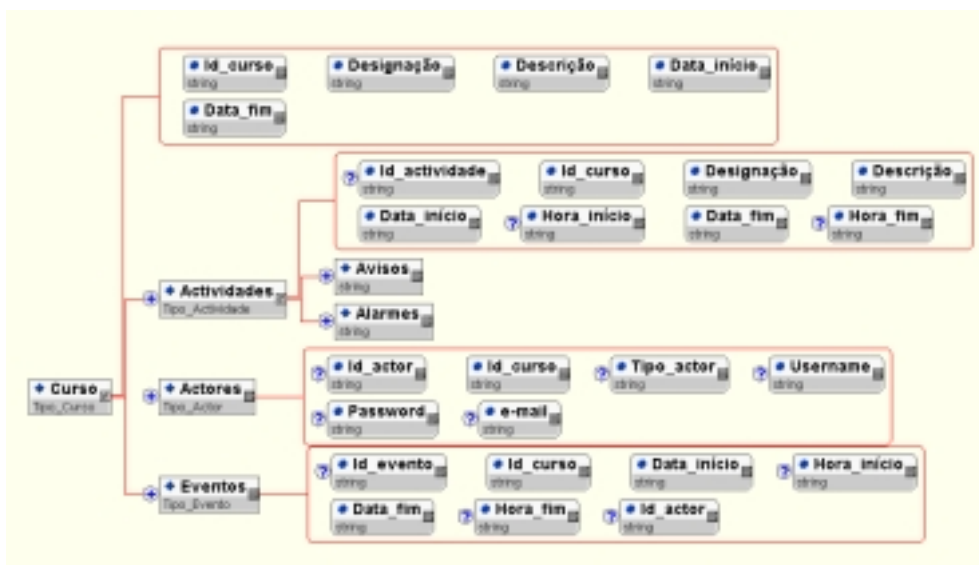
Figura 2 – Unidade elementar correspondente à operacionalização de um curso

Desta fase do trabalho resultou o esquema relacional de uma base de dados capaz de armazenar toda a informação relevante, resultante da interacção entre os diversos intervenientes nas diversas actividades de um curso, bem como a própria caracterização prévia de tal curso.

A validação deste modelo foi feita com base num conjunto alargado de perguntas (questões de competência) que previsivelmente poderão ocorrer num contexto como o referido anteriormente. Uma vez que para cada questão colocada havia sempre um “caminho” na base de dados que levava à obtenção de uma resposta para a mesma, o modelo foi aceite como bom.

As condicionantes de compatibilização, reuso e interoperabilidade impostas pelos trabalhos de standardização em curso, como são os casos do LTSC (LOM), IMS, SIF, SCORM, entre outros, fizeram com que adaptássemos o modelo relacional mencionado, à notação de metadados mais actual e abrangente proposta pelo projecto SCORM. Desta forma estaríamos a desenvolver um trabalho capaz de ser integrado com as normas e pré-normas já existentes, dando assim um contributo para o trabalho global que deverá continuar a desenvolver-se nos tempos mais próximos.

A tradução do modelo E-R referido, para o formato de metadados SCORM, não apresentou grandes dificuldades, podendo mesmo dizer-se que poderia ter sido utilizado um tradutor automático (compilador) para realizar esse trabalho. A figura 3 apresenta o Schema XML (não completo) para o problema que tem vindo a ser descrito



**Figura 3 - Versão gráfica do XML Schema relativo à definição de um curso (Obtida a partir da utilização da ferramenta TurboXML da TIBCO)**

Apresenta-se seguidamente o código XML correspondente a parte do esquema XML da figura 3.

```

<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by Turbo XML 2.3.1.100. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified">
  <xsd:complexType name = "Tipo_Curso">
    <xsd:sequence>
      <xsd:element ref = "Actividades" maxOccurs = "unbounded"/>
      <xsd:element ref = "Actores" maxOccurs = "unbounded"/>
      <xsd:element ref = "Eventos" maxOccurs = "unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "Id_curso" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Designação" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Descrição" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Data_inicio" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Data_fim" use = "required" type = "xsd:string"/>
  </xsd:complexType>
  <xsd:complexType name = "Tipo_Actividade">
    <xsd:sequence>
      <xsd:element name = "Avisos" type = "xsd:string" maxOccurs = "unbounded"/>
      <xsd:element name = "Alarmes" type = "xsd:string" maxOccurs = "unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "Id_actividade" type = "xsd:string"/>
    <xsd:attribute name = "Id_curso" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Designação" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Descrição" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Data_inicio" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Hora_inicio" type = "xsd:string"/>
    <xsd:attribute name = "Data_fim" use = "required" type = "xsd:string"/>
    <xsd:attribute name = "Hora_fim" type = "xsd:string"/>
  </xsd:complexType>
  <xsd:element name = "Actividades" type = "Tipo_Actividade"/>
</xsd:schema>

```

**Figura 4 – Versão XML parcial relativa à definição de um curso  
(Código gerado pela ferramenta TurboXML da TIBCO)**

#### **4. Utilização de Ontologias de Domínio para Modelação da Camada de Gestão**

Dada a não estabilização deste domínio de conhecimento, podemos encontrar inúmeras definições sobre o que é uma ontologia e para o 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 [3].

Borst [4] define 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 [5].

Existem diversos tipos de ontologias – Ontologias gerais, linguísticas e de domínio [6], [7].

Aquelas em que estamos interessados são as 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.

Uma ontologia é o reflexo do mundo real num determinado domínio. Se num caso particular não existir qualquer categorização, a ontologia correspondente deve reflectir tal realidade.

Utilizam-se 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.

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 aplicação em causa.

Noy [8] afirma que uma ontologia é uma descrição formal, 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

São diversas as motivações e as áreas de aplicação. Desde a medicina ao direito, passando pelos sistemas de informação geográficos, temos tido a possibilidade de observar a sua aplicação em quase todas as áreas de conhecimento.

Porque pretenderá alguém desenvolver uma ontologia? Segundo Noy [8], existem várias motivações:

- Para partilhar entendimentos comuns sobre a estrutura da informação, através de pessoas ou agentes de software
- Para permitir o reuso de conhecimento num determinado domínio
- Para tornar explícitas 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 “papéis” ou atributos (slots) e descrição dos valores permitidos para esses slots;
- Preenchimento dos valores dos slots para as instâncias das classes.

Tal como já acontecia com outras abordagens, também agora é aconselhável a enumeração de um conjunto de questões a colocar à ontologia (perguntas de competência) para perceber se a mesma contém ou não a informação necessária [8].

Pode estabelecer-se uma comparação 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.

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 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.

Podemos identificar numa ontologia dois tipos de axiomas – axiomas de derivação e axiomas de consolidação. 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.

Na figura 5 podemos ver uma representação parcial da ontologia de domínio criada para a definição da camada de gestão operacional dos cursos na Web, sendo de realçar as semelhanças existentes entre este modelo, o modelo E-R (não apresentado neste artigo) e a codificação XML da Figura 4.

Com efeito, existem diversas abordagens para a modelação de sistemas, muitas delas apresentando características muito semelhantes. Não há, de facto, 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 os “slots” utilizados 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, que um actor não pode ser simultaneamente do tipo “professor” e “aluno”. Apenas questões de bom senso nos levam a deduzir essa regra.

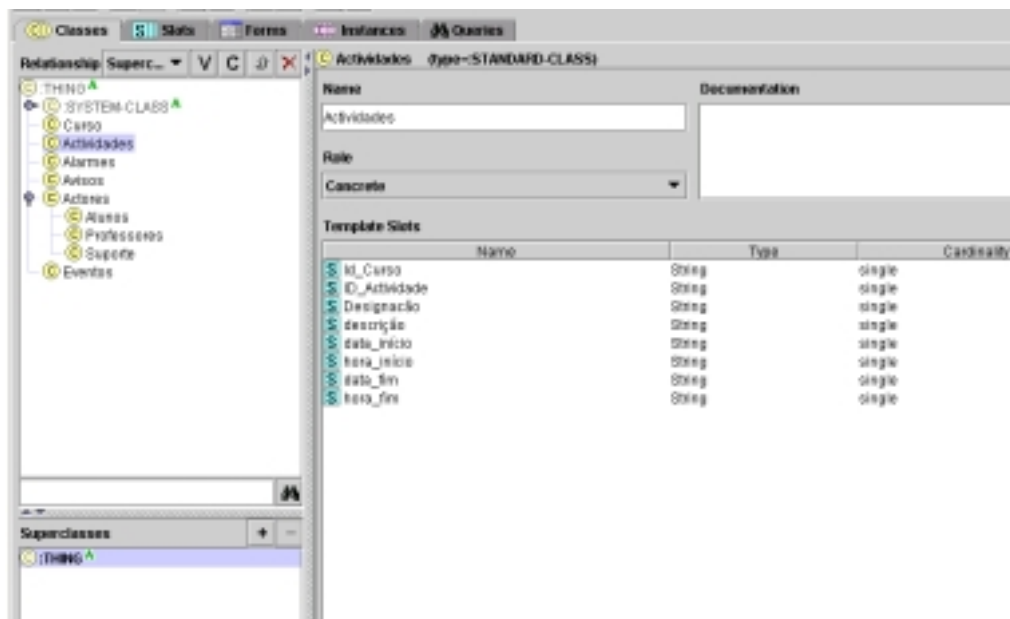
Ao construirmos a ontologia da Figura 5, podemos associar-lhe axiomas que ajudem a estabelecer inequivocamente essa semântica.

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

No caso concreto das ontologias, e do Protégé2000 em particular (ferramenta editora de ontologias utilizada no âmbito deste trabalho), 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 deste axioma poderia ser: Dados dois quaisquer actores a e b, se a é professor e b é aluno, então a tem que ser diferente de b.



**Figura 5 – Vista parcial da ontologia relativa à gestão operacional de cursos na Web (Obtida através da ferramenta Protégé2000)**

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

Apresenta-se na Figura 6 um excerto dos documentos obtidos automaticamente com base na ontologia da Figura 5, relativo à classe Actores do modelo que temos vindo a referir.

A título de curiosidade, referiremos a possibilidade de gerar um ficheiro com base na ontologia em formato XMI, capaz de ser lido por variadas ferramentas, e de poder dar origem, de uma forma automática, aos correspondentes diagramas de classes UML.



**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
<i>Password</i>		String		0:1	
<i><u>id_actor</u></i>		String		0:1	
<i>Username</i>		String		0:1	
<i>e-mail</i>		String		0:1	
<i><u>id_curso</u></i>		String		0:1	

**Figura 6 – Metadados referentes à ontologia de gestão operacional de cursos na Web (Obtida através da ferramenta Protégé2000)**

## 5. Conclusões

Um longo caminho foi percorrido já, desde a identificação da necessidade da especificação da camada de gestão operacional de cursos suportados por plataformas de e-Learning, como forma de de suprir estes sistemas de capacidades de reação em tempo real às situações anómalas verificadas durante a operacionalização dos cursos. Tais situações, pela experiência adquirida, são causadoras de perturbação no normal funcionamento desses cursos.

Observando-se o trabalho que tem vindo a ser realizado pelas organizações mais importantes do sector, verifica-se que esta componente continua a não ser objecto da preocupação de tais organizações e, portanto, uma solução para o problema apresenta-se como necessária e oportuna.

Depois de caracterizado o sub-sistema em causa, temos vindo a fazer uma evolução no sentido de tentar encontrar a melhor abordagem para a escrita dos metadados de gestão que, contudo, terão que cumprir dois objectivos. Por um lado deverão constituir uma verdadeira solução para o problema. Por outro, terão que obedecer ou estar em concordância com os trabalhos de normalização já realizados e em curso, relativos às outras camadas da arquitectura global de uma plataforma de e-Learning. Estamos especialmente preocupados em manter uma sintonia estreita com o trabalho realizado no âmbito do projecto SCORM.

Pela facilidade de utilização na modelação de sistemas, pelo formalismo que introduzem, pelas possibilidades de derivação de versões em linguagem Web standard, como é o caso do XML, parece-nos que as ontologias de domínio são a ferramenta acertada para nos apoiarmos no restante percurso que ainda temos que fazer.

## Referências bibliográficas

- [1] SCORM (2004), ADL Sharable Content Object Reference Model, Version 1.3. 2004 Overview. <http://www.adlnet.org> [último acesso a 22 de Fevereiro de 2004]
- [2] ODL (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).
- [3] Ramalho, J. C., Librelotto, G. R., Henriques, P. R. (2003). *Ontology driven Websites Metamorphosis: a framework to specify and manage ontology driven websites*, in S. Costa, J. Carvalho, A. Baptista e A. Moreira (eds.), Proceedings of the 7<sup>th</sup> ICCC/IFIP International Conference on Electronic Publishing. Guimarães, Universidade do Minho.
- [4] Borst, W. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Ph.D. thesis, University of Twente, Enschede.
- [5] Studer, R., Benjamins, V. & Fensel, D (1998). *Tknowledge Engineering, principles and methods*. Data and Knowledge Engineering, vol.25, nº1-2, pp. 161-197.
- [6] van Heijst, G., Schreiber, A. & Wielinga, B. (1997). *Using explicit ontologies in KBS development*. International Journal of Human-Computer Studies, vol.46, nº 2/3, pp. 183-192.
- [7] 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, NL.
- [8] Noy, N. F., McGuinness, D. L. *Ontology Development 101: A Guide to Creating Your First Ontology*.  
[http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf) [último acesso a 20 de Maio de 2004].