

# *PROJETO E DESENVOLVIMENTO DE UM SISTEMA MULTI-AGENTES PARA OBJETOS INTELIGENTES DE APRENDIZAGEM BASEADO NO PADRÃO SCORM*

*Júlia Marques Carvalho da Silva*  
Centro de Ciências Tecnológicas  
da Terra e do Mar,  
Universidade do Vale do Itajaí  
88302-202 - Itajaí - SC - Brasil  
julia@univali.br

*Natanael Bavaresco*  
Departamento Informática e Estatística,  
Universidade Federal de Santa Catarina  
88040-900 - Florianópolis - SC - Brasil  
natanael@inf.ufsc.br

*Ricardo Azambuja Silveira*  
Departamento Informática e Estatística,  
Universidade Federal de Santa Catarina  
88040-900 - Florianópolis - SC - Brasil  
silveira@inf.ufsc.br

---

**Resumo:** Em sistemas de aprendizagem existem dois aspectos que devem ser considerados: (i) a adaptabilidade, que diz respeito às diferentes necessidades e estilos de aprendizagem dos alunos; e (ii) a reusabilidade, que visa apoiar a confecção dos cursos. Acredita-se que estas características podem ser alcançadas através da interligação entre Sistemas Multiagentes (SMA) e Objetos de Aprendizagem (Learning Objects – LO), resultando na abordagem denominada de Objetos Inteligentes de Aprendizagem (Intelligent Learning Objects – ILO). Este trabalho apresenta a proposta de um SMA para ILO baseada no padrão SCORM, utilizando a Engenharia de Sistemas Multiagentes (Multiagent System Engineering - MaSE).

---

**Palavras-chave:** Objetos de Aprendizagem, Sistemas Multiagentes, SCORM.

---

**Abstract:** There are two aspects that must be considered in learning systems: adaptability and reusability. Adaptability consists in to be flexible according to different contexts and learning styles of students. Reusability aims to support courses design. These goals can be reached by joining Multi-agent Systems and Learning Objects technologies. The result consists on Intelligent Learning Objects (ILO), whose goal is to promote richer learning experiences and to provide reusability and adaptability of these objects in a more effective way. This paper describes the design and the specification of ontology to ILO, based on SCORM reference model, using Multiagent System Engineering (MaSE).

---

**Keywords:** Learning Objects, Multiagent Systems, SCORM.

## 1. INTRODUÇÃO

Este trabalho descreve como a adaptabilidade e reusabilidade de objetos de aprendizagem podem ser alcançadas através da aplicação de ambientes de aprendizagem baseados em Sistemas Multi-agentes em conformidade com as especificações do modelo SCORM. Para tanto, é apresentada a modelagem formal de uma abordagem proposta por [13], na qual objetos de aprendizagem são construídos com base no paradigma de agentes. A fundamentação tecnológica desta abordagem é constituída por uma integração entre tecnologias

desenvolvidas para Objetos de Aprendizagem e para Sistemas Multi-agentes (SMAs).

O conceito central apresentado é o de Objeto Inteligente de Aprendizagem, entidade que corresponde a um agente com a capacidade de gerar experiências de aprendizagem reutilizáveis, no mesmo sentido que os objetos de aprendizagem. Para isto, é apresentado um modelo formal de uma sociedade multi-agente capaz de sustentar tal conceito.

Os Objetos Inteligentes de Aprendizagem, aqui descritos, comunicam-se com o repositório de objetos e o

sistema gerenciador de aprendizagem, todos eles implementados na forma de agentes, os quais compreendem um ambiente de aprendizagem. Os três papéis se complementam sendo capaz de promover as experiências de aprendizagem. A estratégia pedagógica será resultante da interação que ocorre entre estas entidades.

Conforme [3], os agentes inteligentes oferecem uma grande contribuição para a implementação de ambientes colaborativos, pois podem se comunicar em qualquer rede baseada nos protocolos. Esta característica faz com que a cooperação possa ocorrer tanto em pequenos laboratórios de aula como através da Internet, sem limite de distância.

## 2. OBJETOS INTELIGENTES DE APRENDIZAGEM

Um Objeto de Aprendizagem é uma peça de software que proporciona algum tipo de experiência de aprendizagem, e pode ser aplicado ao aprendizado em diversos momentos, e em diferentes cursos e situações [5], [10] e [14]. Os objetos podem ser controlados por algum Sistema Gerenciador de Aprendizagem (*Learning Management Systems - LMS*), contudo estão limitados a funcionar de uma forma restrita.

[13] apresentam uma abordagem sobre os Objetos de Aprendizagem, agregando a tecnologia de Sistemas Multi-agentes (SMAs), denominado Objetos Inteligentes de Aprendizagem (*Intelligent Learning Objects - ILO*). A utilização de SMAs amplia as possibilidades dos ambientes de aprendizagem tradicionais, por apresentarem características como [2]:

- Um agente é um pedaço de software que trabalha de forma contínua e autônoma, dentro de um ambiente;
- Um agente pode interferir em um ambiente de forma flexível e inteligente, sem necessitar de intervenção humana; e,
- Um agente pode se comunicar com outros agentes através de troca de mensagens, usando Linguagens de Comunicação entre Agentes (*Agent Communication Language – ACL*).

As características acima citadas podem ser aplicadas ao contexto dos Objetos Inteligentes de Aprendizagem, da seguinte forma: um ILO é uma peça de software, com um comportamento autônomo, cujo comportamento é adaptativo de acordo com sua percepção do meio ambiente e das informações recebidas de outros agentes, devido a sua capacidade de inferência, sem necessitar de intervenção externa contínua. A Figura 1 representa a união dos conceitos explicitados que dão origem a um objeto inteligente de aprendizagem.

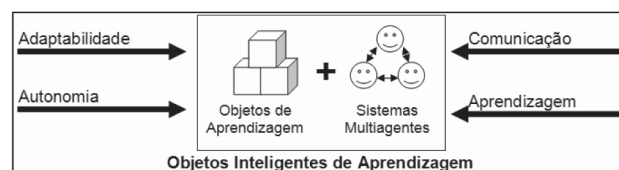


Figura 1. Objetos Inteligentes de Aprendizagem

Uma possibilidade interessante refere-se à capacidade de aprendizagem, na qual os agentes podem implementar, baseada ou não em alguma técnica de Inteligência Artificial. Logo, um objeto de aprendizagem com esta característica poderá adquirir novos conhecimentos e comportamentos no decorrer de sua existência, através da interação com os alunos e, até mesmo, com outros objetos de aprendizagem, tais como: adquirir informações sobre os alunos como as suas preferências e estilos cognitivos e verificar quais as melhores estratégias de aprendizagem em cooperação com os demais agentes.

## 3. SCORM

O SCORM (*Sharable Content Object Reference Model*) é um modelo de referência para construção de objetos de aprendizagem de modo que eles sejam reutilizáveis e interoperáveis para qualquer LMS, desde que suporte este modelo, isto é, padronizando uma maneira única de iniciar e comunicá-los com o LMS. Para que objeto de aprendizagem torne-se um objeto SCORM, deve atender os seguintes requisitos [1]:

- Reusabilidade: deve ser modificado facilmente e usado por diferentes ferramentas de desenvolvimento e plataformas, além de ser aplicável em múltiplos contextos;
- Acessibilidade: capacidade de ser encontrado e torná-lo disponível se possível por aprendizes e desenvolvedores de conteúdos, de qualquer local remoto;
- Interoperabilidade: ser operável em diversos tipos de hardware, sistemas operacionais e navegadores web; e
- Durabilidade: não deve ser necessário realizar modificações significativas (reconfigurar, reimplementação) com novas versões de software.

Ainda, o SCORM apresenta um conjunto de especificações que abrange: (i) visão geral, (ii) modelo de agregação de conteúdo e (iii) ambiente de execução (*Run-Time*). O modelo de agregação de conteúdo é responsável por promover um conceito comum para a construção de conteúdos de aprendizagem, sendo possível descobrir, reutilizar, compartilhar e o interoperar. Ele também define que conteúdos de aprendizagem podem ser identificados e descritos, agregando em um curso ou em parte de um curso, além de movê-lo entre LMS e repositórios. O modelo inclui especificações para agregar conteúdo e definir metadados. Ainda, são definidos os seguintes tipos de componentes: Recursos (*Assets*), Objeto de Conteúdo Compartilhado (SCO), Atividades, Organização de Conteúdo e Agregação de Conteúdo (*Content Aggregation*).

Já o ambiente de execução é responsável pela definição da implementação, isto é, como o objeto de aprendizagem deve se comunicar com o LMS. Conforme a Figura 2, para que isto seja possível, é necessária a utilização de um adaptador da API SCORM, denominado APIAdapter [1].

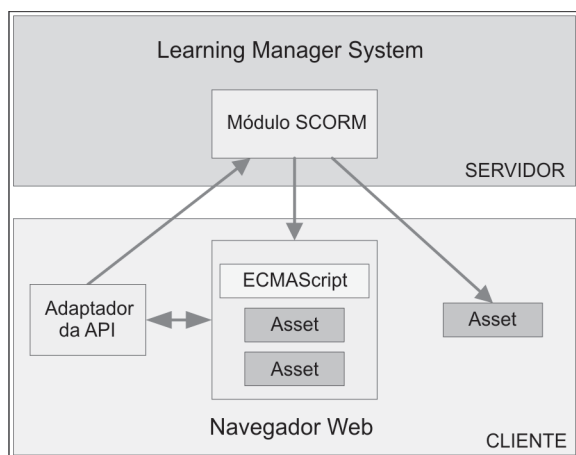


Figura 2: Ambiente de Run-Time

O uso de uma API comum provê uma forma de padronização para que os SCOs se comuniquem com o LMS, encapsulando detalhes de programação nem sempre interessantes ao contêudista. Em termos gerais, uma API contém um conjunto de funções pré-definidas que o SCO pode acessar quando estiver ativo. Já um adaptador de API é uma biblioteca

de funcionalidades que podem ser manipuladas externamente. Logo, o LMS apenas precisa implementar este adaptador de forma a permitir que o objeto tenha acesso a informações que possa vir utilizar, por exemplo, o nome do aluno ou em que parte do objeto o aluno parou na última vez que o acessou. As funções disponíveis na API são apresentadas na Tabela 1.

A comunicação do SCO com o LMS através de uma instância do adaptador da API atravessa por diversos estados, conforme ilustra a Figura 3. Os estados do adaptador da API especificam as respostas fornecidas dada a ocorrência de um evento. Durante cada um destes estados há um conjunto de diferentes atividades que o SCO pode realizar. Os estados transpassados pela API são: “Não Iniciado”, “Iniciado” e “Encerrado”.

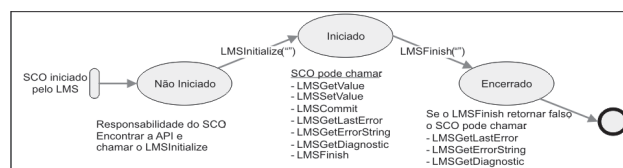


Figura 3: Estados de transições da API

Tabela 1: Funções da APIAdapter

Classificação	Função	Descrição
Estado de Execução	LMSInitialize	Indica ao adaptador da API que o SCO irá se comunicar com o LMS.
	LMSFinish	O SCO deve chamar esta função quando não precisará mais se comunicar com o LMS.
Transferência de Dados	LMSGetValue	Permite que o SCO obtenha uma informação do LMS.
	LMSSetValue	Permite que o SCO envie uma informação ao LMS.
	LMSCommit	Permite que o SCO solicite ao LMS que registre os valores enviado a ele de forma persistente.
Gerenciamento do Estado	LMSGetLastError	Possibilita ao SCO saber se alguma operação falhou ou não.
	LMSGetErrorString	Retorna uma mensagem textual sobre o erro obtido.
	LMSGetDiagnostic	Fornece informações adicionais sobre um determinado erro corrente.

#### 4. SOCIEDADE DE AGENTES PARA ILO

Para o desenvolvimento dos objetos de aprendizagem sob a perspectiva de agentes, foi necessária a modelagem de uma sociedade de agentes, na qual contemple as entidades fundamentais no processo de ensino- aprendizagem: conteúdo – ambiente – aluno. Sendo assim, foram identificados os seguintes tipos de agentes:

- **Agente LMS:** representa os sistemas de gerenciamento de aprendizagem, lidando com os aspectos administrativos e gerenciais que envolvem os ambientes de aprendizagem. Além disso, provê o acesso dos aprendizes aos ILOs, e fornece informações do aprendiz aos ILOs;
- **Agente ILO:** encapsula os objetos de aprendizagem, sendo capaz de gerar experiências de aprendizagem aos alunos;

- **Agente ILOR (Intelligent Learning Object Repository):** são abstrações dos sistemas de repositório de objetos de aprendizagem, armazenando dados sobre os ILOs, possibilitando a usuários ou agentes encontrá-los; e,
- **Agente Learner:** contempla a representação do aluno perante o ambiente e seus objetos. É responsável pelas operações de armazenamento e resgate de informações sobre o modelo de dados do aluno.

O SMA proposto é apresentado na Figura 4, onde se ilustra a dinâmica dos agentes descritos acima. Inicialmente, os aprendizes interagem com o agente LMS a fim de obter experiências de aprendizagem. O agente LMS busca o ILO mais apropriado para a necessidade de aprendizagem do aprendiz e o invoca. O ILO é então responsável por gerar as experiências de aprendizagem ao aprendiz. Nesta tarefa, é possível comunicar-se com o

agente LMS e com outros agentes ILO a fim de promover uma experiência de aprendizagem mais rica. Toda comunicação é promovida pela troca de mensagens no padrão FIPA-ACL. O ambiente onde tais agentes habitam estão no padrão FIPA, o qual provê todo o mecanismo necessário para troca de mensagens entre os agentes..

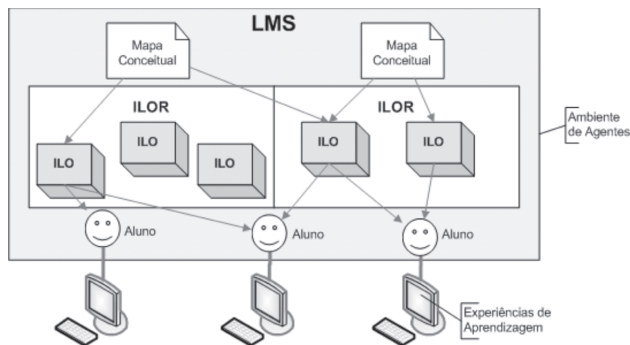


Figura 4: Sistema multi-agentes proposto por Silveira et al. (2005)

## 5. MODELANDO O SISTEMA MULTI-AGENTE

O Sistema Multiagente foi modelado através da metodologia proposta por [4], denominada MaSE. Ela permite o desenvolvimento de sistemas multiagentes com base nos princípios da Engenharia de Software. Para tanto, o processo de desenvolvimento é dividido em duas fases principais: a análise e a modelagem; onde cada um provê um conjunto de etapas a serem seguidas. A Figura 5 apresenta a metodologia MaSE.

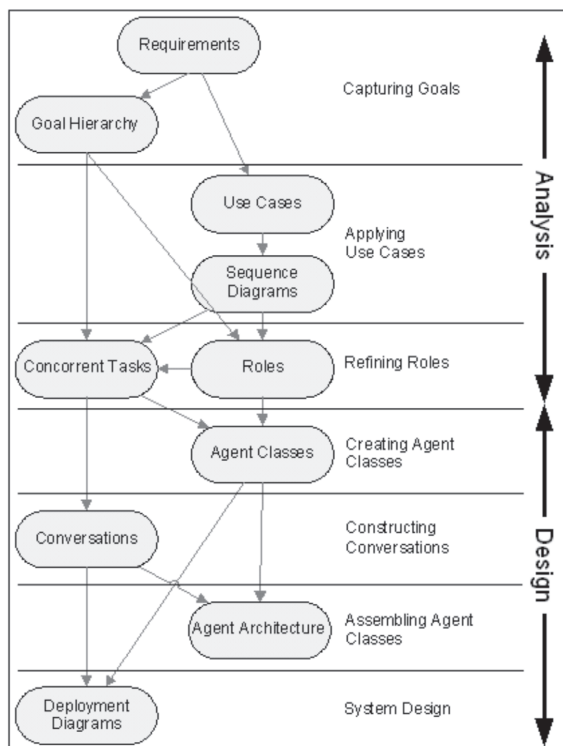


Figura 5: Diagrama de hierarquia de objetivos DeLoach e Wood (2001)

A fase de análise consiste nas seguintes etapas: capturar dos objetivos, estabelecer os casos de uso, e definir os papéis; já a fase de modelagem contempla as etapas: criar as classes de agentes, desenvolver os diálogos, reunir as classes dos agentes, desenvolver a estrutura do sistema.

A metodologia MaSE foi desenvolvida para ser aplicada de forma interativa. Espera-se que o desenvolver possa navegar entre as etapas em diversos momentos, acarretando em um modelo completo e consistente [4]. Os modelos são transformados em diagramas que permite descrever os agentes, seus diálogos e estrutura interna, em diferentes níveis de detalhamento. Ainda, o desenvolvedor pode selecionar quais diagramas irá utilizar e em que ordem eles serão descritos. Ao final, obtém-se uma consistência entre os modelos do MaSE de forma a satisfazer os objetivos iniciais do sistema.

Na seqüência é apresentada a modelagem da sociedade de agentes proposta por [13], utilizando a ferramenta agentTool, na qual implementa a metodologia MaSE através de seus diagramas.

### 5.1. CAPTURANDO OS OBJETIVOS

A identificação dos objetivos do SMA visa conhecer as especificações do sistema e transformá-las em uma estrutura de conjunto de objetivos. Conforme a metodologia, é necessário conhecer qual é o objetivo principal do SMA, para então expandir em sub-objetivos os quais permitirão atingir o primeiro. No caso dos Objetos Inteligentes de Aprendizagem, o objetivo fundamental é proporcionar aos aprendizes experiências de aprendizagem. Para que isto seja possível, é necessário que as três entidades (ambiente, objeto e aluno) estejam aptos a realizar suas tarefas específicas, como, por exemplo, o ambiente deve ser capaz de receber uma solicitação de experiência de aprendizagem. Os objetivos identificados para este contexto são apresentados na Figura 6.

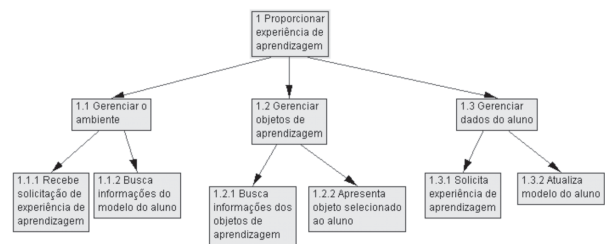


Figura 6: Diagrama de hierarquia de objetivos

### 5.2. ESTABELECEENDO OS PAPÉIS

Esta etapa permite definir através de um diagrama os papéis (representados em retângulos), tarefas (representadas pelas elipses) e protocolos de comunicação (representados pelas setas) que o SMA deverá conter. No diagrama, todos os objetivos estabelecidos inicialmente devem ser vinculados a um papel, conforme é apresentado na Figura 7.

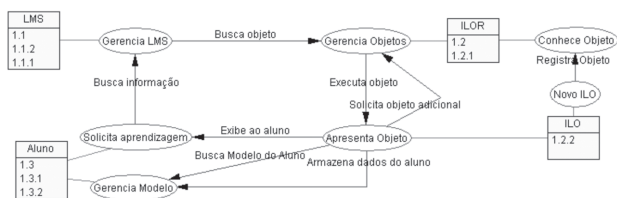


Figura 7: Diagrama de hierarquia de papéis

Neste trabalho foram identificados quatro papéis que permitirão prover uma experiência de aprendizagem:

- **Aluno:** gerenciar os dados dos alunos, solicitar nova experiência de aprendizagem e atualizar o modelo do aluno;
- **LMS:** gerenciar o ambiente, receber solicitação de experiências de aprendizagem, buscar informações sobre o aluno;
- **ILOR:** gerenciar os objetos de aprendizagem (em seu repositório) e buscar um objeto de aprendizagem; e,
- **ILO:** apresentar o conteúdo do objeto selecionado ao aluno.

Tais papéis permitirão que as seguintes tarefas sejam desenvolvidas, utilizando um determinado protocolo:

- **Solicita experiência de aprendizagem:** informa ao LMS que deseja aprender algo;
- **Gerencia o modelo do aluno:** responsável pelo armazenamento e recuperação de informações referente ao aluno;
- **Gerencia o LMS:** busca novos objetos para disponibilizar ao aluno;
- **Gerencia os Objetos:** solicita que um objeto seja executado;
- **Apresenta o Objeto:** apresenta o objeto ao aluno, e caso a experiência de aprendizagem necessite de complemento com o uso de outros objetos, os solicita;
- **Registra o ILO:** apresenta-se ao repositório como um novo objeto; e,
- **Conhece o objeto:** carrega os metadados do objeto no ambiente.

### 5.3. DESENVOLVENDO AS CLASSES DOS AGENTES

As classes de agentes são identificadas a partir dos papéis (caixas), e os diálogos (linhas), que conectam as classes. O diagrama das classes de agentes para Objetos Inteligentes de Aprendizagem é mostrado na Figura 8.

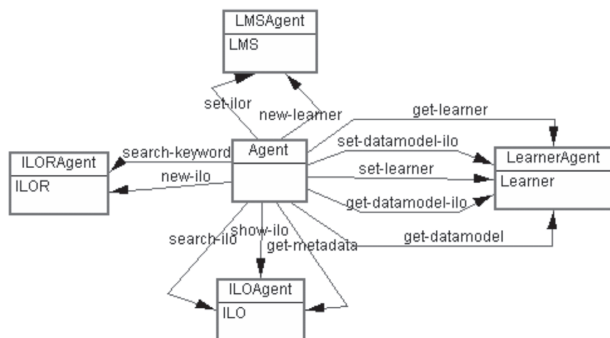


Figura 8: Diagrama de classes de agentes

Estabeleceram-se três classes de agentes que representarão os papéis de agente LMS, agente ILOR e agente ILO, no que diz respeito a retornar uma determinada informação solicitada. Ainda, conforme descrito na sociedade, qualquer agente poderá estabelecer diálogo em busca de alguma ação, desta forma foi desenvolvido um agente genérico que englobará todos os papéis, representando esta generalidade.

### 5.4. DIÁLOGOS ENTRE OS AGENTES

O diagrama de diálogo entre agentes é formado por um par de estados finitos que define a comunicação entre duas classes de agentes exclusivamente. O inicializador começa o diálogo enviando a primeira mensagem. O final do diálogo é representado pelo estado finito terminal.

Conforme identificado no diagrama de classes de agentes, existem doze diálogos referentes à ontologia “ilo-ontology”, desenvolvida para a aplicação no domínio de Objetos Inteligentes de Aprendizagem.

Os diálogos descritos pela ilo-ontology seguem os protocolos da especificação FIPA [7]. Por exemplo, o fluxo de mensagens para o diálogo get-metadata (Figura 9) baseia-se no documento SC00026H – FIPA Request Interaction Protocol Specification.

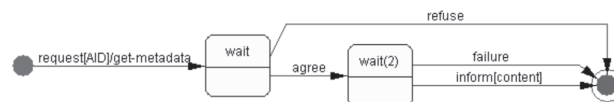


Figura 9: Diagrama do diálogo get-metadata

O diálogo é composto pelas seguintes fases: (i) um agente envia uma requisição ao agente ILO informando seu identificador e o identificador do diálogo “get-metadata”; (ii) o agente ILO informa ao agente se aceita ou se rejeita a requisição; (iii) o agente ILO retorna a informação solicitada (metadados) ou se o diálogo falhou.

#### 5.4.1. ONTOLOGIA ILO-ONTOLOGY

Conforme a [7], ontologia é um conjunto de conceitos de um domínio específico. Geralmente, uma ontologia FIPA é definida pelo uso de predicados, ações e conceitos. Embora simples esta definição, é muito pragmática e satisfaz todas as requisições de um processo de comunicação entre agentes definida na pesquisa.

Para o desenvolvimento da ontologia, foi utilizada a ferramenta Protégé [12], a qual possui uma expressiva aceitação na comunidade científica, além de disponibilizar mecanismos de integração com framework selecionado para o desenvolvimento dos agentes. Os conceitos do LMS, ILO, ILOR e do aluno são ilustrados na Figura 10. Nota-se que há uma interligação semântica entre os conceitos: o modelo de dados do aluno se refere à interligação do desempenho obtido pelo aluno em um determinado objeto. Já o ILOR implementa uma lista de objetos que ele possui.

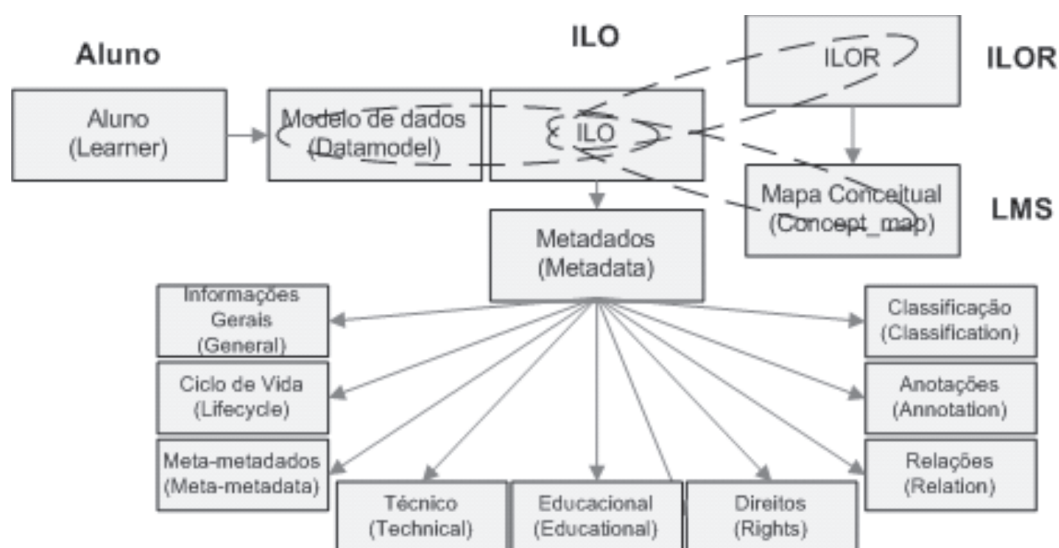


Figura 10: Conceitos da Ontologia.

Outra entidade identificada é relativo o mapa conceitual. Percebeu-se durante a pesquisa, a necessidade de uma estrutura que norteasse o desenvolvimento de um curso. Um mapa conceitual permite que sejam identificados os conceitos a serem abordados, e relacioná-los através de pré-requisitos conceituais e temporais. Na arquitetura proposta, cada objeto inserido no SMA deverá ser relacionado a um ou mais conceitos, para que no momento de sua adaptação eles sejam identificados corretamente, evitando o problema de conceitos de mesmo nome porém de semânticas diferentes. Por exemplo, o termo “janela” pode-se referir tanto a um elemento de uma obra / construção, quanto a uma interface de interação entre um software e um usuário. Ao inserir estes conceitos dentro de um mapa, pode-se distinguir seus contextos, oferecendo ao aluno um objeto semanticamente correto.

A ontologia contém meta informações sobre o LMS, o aluno, o objeto de aprendizagem e seu repositório. Os dados contidos referentes ao aluno incluem um identificador, seu nome e uma referência para o seu modelo de dados, o qual será responsável pelo armazenamento de sua aprendizagem ao utilizar cada objeto de aprendizagem. Já sobre os objetos de aprendizagem, tem-se um conjunto de informações que o caracterizam. A definição de quais variáveis seriam relevantes para se conhecer um objeto, baseou-se no padrão IEEE 1484.12.1-2002 - *Learning Object Metadata* (2002) [8], o qual incluem a relação entre o objeto e o aproveitamento do aluno.

O repositório dos objetos apresenta uma lista responsável por armazenar as referências dos objetos ativos, além de se conhecer o local onde tal repositório está localizado. Por fim, o LMS deve conhecer seus alunos, repositórios de objetos, e os mapas conceituais que definirão cada um dos cursos disponíveis no ambiente. O mapa conceitual estabelece uma seqüência em que os conceitos devem ser abordados, permitindo ainda que o professor, ao montar seu curso, defina qual(is) objeto(s)

ele prefere que o curso aborde ao aluno. O conceito de mapa conceitual aqui apresentado difere da seqüência dos *assets* contidos em um objeto SCORM.

## 6. DESENVOLVIMENTO DO FRAMEWORK

A arquitetura apresentada na Figura 11 representa três perspectivas que são detalhadas nas subseções a seguir. A seqüência foi delineada conforme o conhecimento necessário para a compreensão da arquitetura. Logo, a primeira relaciona o uso dos recursos tecnológicos para o desenvolvimento da aplicação. A segunda indica a seqüência do funcionamento da arquitetura. A terceira parte especifica a implementação de fato dos agentes que habitam o SMA.

Sob o ponto de vista tecnológico, a arquitetura baseia-se no paradigma cliente-servidor. O lado servidor contém a plataforma JADE [9] que gerencia os agentes, ontologias e regras de negócio; além de utilizar uma base de dados que realizará a persistência dos dados, facilitando as consultas para construção da estrutura da plataforma. Para a implementação do protótipo, foi utilizado o sistema gerenciador de banco de dados PostgreSQL e desenvolvido um modelo de dados com as mínimas informações utilizadas pelo SMA, a fim de simular a existência de um LMS.

Já do lado cliente consta o objeto de aprendizagem, o qual pode conter qualquer tipo de mídia que possa ser apresentada através de uma página HTML, além da estrutura projetada pelo padrão o qual foi desenvolvido.

Para que estas duas estruturas se comuniquem foi utilizada a tecnologia RMI promovida pela linguagem Java. Isto permite que independentemente do LMS utilizado para execução dos objetos, é possível se conectar ao SMA, desde que implemente a API do modelo SCORM. Para que isto ocorra, é necessário apenas modificar: a página HTML de acesso ao objeto, incluindo um applet não visível ao aluno que se conectará ao agente, que por sua vez implementará a API do padrão SCORM, que irá invocar a conexão com o SMA.

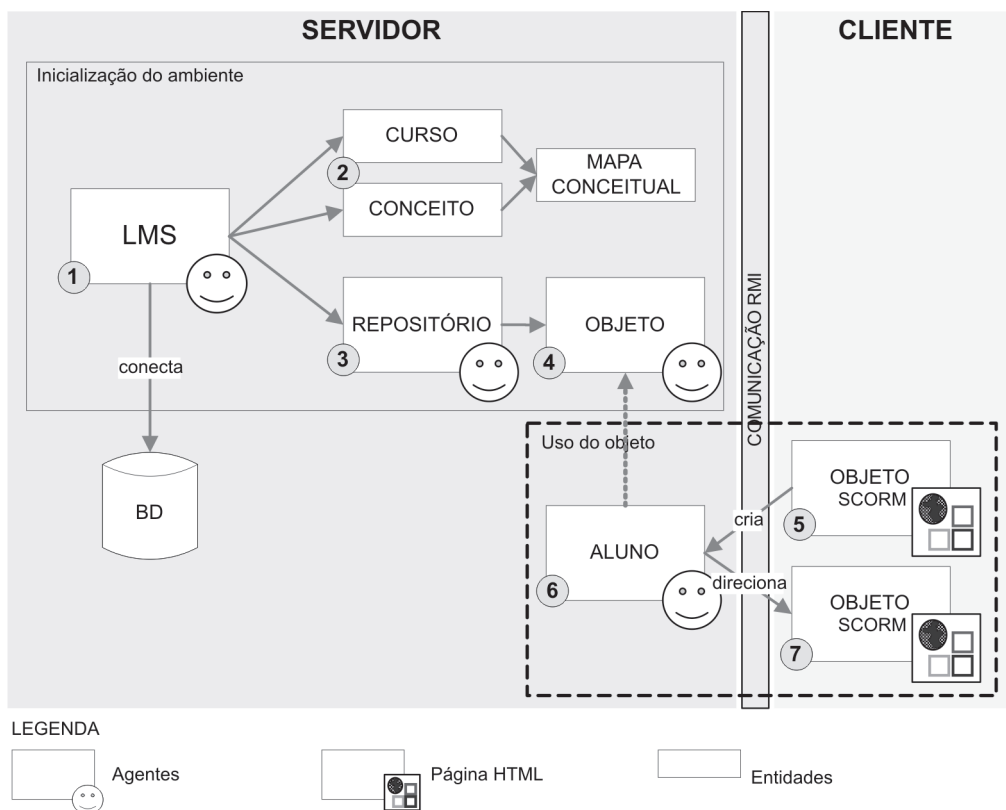


Figura 11: Arquitetura do framework proposto.

Retomando a arquitetura ilustrada na Figura 11, percebe-se a existência de círculos com números. Eles representam a seqüência do ciclo de vida do SMA, ou seja, quais são os passos a serem seguidos para seu funcionamento pleno:

1. Preparação do ambiente através do agente LMS: Para que a estrutura funcione, faz-se necessária a preparação do ambiente por meio da execução do servidor JADE, de registro e servidor serviços RMI. Juntamente a inicialização da plataforma JADE, é criada uma instância do agente LMS, responsável pelo gerenciamento da estrutura interna do SMA.
2. Criação das entidades de apoio (cursos, conceitos e mapas): O agente LMS verifica na base de dados do LMS os cursos existentes, estabelecendo os mapas e conceitos que são abordados. Estas entidades apoiam a tomada de decisão através da entidade de estratégia de ensino abordada.
3. Criação dos agentes que representam os repositórios de objetos de aprendizagem: Na seqüência, o agente LMS verifica quais são os repositórios de objetos de aprendizagem que o LMS trabalha e então cria sua representação através de agentes.
4. Criação dos agentes que representam objetos de aprendizagem: A partir da identificação dos repositórios, são verificados os objetos existentes e então criados seus agentes. Os agentes terão as informações carregadas através do arquivo de metadados que especificam o conteúdo dos objetos.

Os passos descritos até o momento dizem respeito a estrutura de serviços que devem ser estabelecidos antes da execução dos objetos. A seguir são apresentadas as etapas relacionadas ao fluxo de exibição dos objetos de aprendizagem, o qual pode ocorrer diversas vezes, conforme a solicitação de apresentação dos conteúdos.

5. Inicialização do objeto de aprendizagem através do LMS: Através de um LMS, o aluno solicita o acesso a um objeto de aprendizagem.
6. Criação do agente que representa o aluno utilizando o objeto: Juntamente com a etapa anterior, também é estabelecida conexão com a plataforma JADE, onde é criado um agente que representa o ato do aluno visualizar o conteúdo através do agente LMS.
7. Redirecionando a outro objeto: Durante a execução do objeto de aprendizagem selecionado pelo aluno, o SMA pode perceber a necessidade de complemento da aprendizagem. Nesse momento, pode ser encaminhado um outro objeto a esse aluno. Cabe ressaltar que esta ação não é obrigatória, ela ocorre conforme a necessidade individual do aluno.

Sobre a especificação do comportamento de cada um dos agentes desenvolvidos na arquitetura, tal como os diálogos estabelecidos entre eles, mostra-se uma descrição detalhada na seção seguinte..

#### 6.1. DESENVOLVIMENTO DO FRAMEWORK

Conforme descrito na modelagem, foram desenvolvidos quatro agentes para atender as necessidades



linguagem de programação utilizada pelo LMS. Isto se deve ao fato da independência proporcionada através da utilização de comunicação via RMI, ou seja, as regras de negócio e estrutura do ambiente não são afetadas. Deve-se apenas conhecer a implementação do adaptador de API do LMS e adicionar a este, as chamadas aos métodos da classe ClientApplet.

## 8. CONCLUSÕES

O conceito de Objetos Inteligentes de Aprendizagem proposto por [13] foi modelado utilizando uma metodologia própria para sistemas multi-agentes, baseada em conceitos da engenharia de software. Esta metodologia facilita o entendimento e possibilita focar o real objetivo do sistema que se deseja construir. Devido à interligação entre os modelos, é possível estabelecer vínculos em todas as etapas, garantindo que todos os objetivos identificados sejam atendidos pelos agentes e diálogos entre eles.

A modelagem de um sistema já proposto permitiu identificar pontos que devem ser modificados ou ampliados. Foi necessária a redefinição dos agentes, e seus respectivos papéis. A comunicação entre agentes foi revista, onde foram propostos novos diálogos e adequados aos protocolos de comunicação FIPA.

Acredita-se que dentre os benefícios em utilizar sistemas multi-agentes aliado ao paradigma de objetos de aprendizagem, refere-se a possibilidade de gerar seqüenciamentos dinâmicos. Isto é, conforme a interação do aluno com o objeto, que ele seja capaz de identificar o modelo do aluno e direcioná-lo a um conteúdo que possa ampliar seus conhecimentos. Entretanto, sabe-se que esta tarefa não é considerada trivial, nem solucionável em um único trabalho.

Sendo assim, o presente trabalho buscou formalizar a integração dos objetos a um SMA, de forma a promover um framework consistente. A partir de então, espera-se que seja possível a inclusão de tomadas de decisão sobre como e quando o seqüenciamento deve ocorrer.

Como trabalhos futuros, a nova edição do padrão SCORM [1] incluiu especificações sobre seqüenciamento e navegação entre os conteúdos. A arquitetura aqui proposta não estabelece detalhes sobre sua aplicabilidade a este novo contexto. Portanto, um trabalho futuro pode ser desenvolvido neste sentido, a fim de adequar o SMA desenvolvido conforme as essas especificações.

Aliado a este, sugere-se um estudo sobre a seleção dos conteúdos de aprendizagem que o aluno deve ser orientado, seja aplicando uma técnica de inteligência artificial ou negociação entre agentes, por se tratar de um framework aplicado em um SMA.

## REFERÊNCIAS

- [1] ADL. *The SCORM Run-Time Environment*. Advanced Distributed Learning, 2001.
- [2] Bradshaw, J. M.. An introduction to software agents. Bradshaw, J. M. Ed. *Software Agents*. Massachusetts: MIT Press, 1997.
- [3] Costa, M. T. C. *Uma Arquitetura Baseada em Agentes para Suporte ao Ensino a Distância*. Doutorado em Engenharia da Produção. Universidade Federal de Santa Catarina, 1999.
- [4] DeLoach, S. A.; Wood, M. *Developing Multiagent Systems with agentTool*. Proceedings of Lecture Notes in Artificial Intelligence. Springer – Verlag. Berling, 2001.
- [5] Downes, S. *Learning objects: resources for distance education worldwide*. International Review of Research in Open and Distance Learning, 2(1). 2001.
- [6] EXE. eLearning XHTML Editor. Disponível em: <<http://exelearning.org>>. Acesso em: 29 set. 2007.
- [7] FIPA: The foundation for Intelligent Physical Agents: Specifications (2002). Disponível em: <http://www.fipa.org>. Acessado em: 7 de Julho de 2005.
- [8] IEEE (2002). Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002. Disponível em: <http://ltsc.ieee.org>. Acessado em: 10 de Janeiro de 2007.
- [9] JADE. Java Agent Development Framework. Disponível em: <http://jade.tilab.com>, Acessado em: 25 de Junho de 2007.
- [10] Mohan, P.; Brooks, C. , *Engineering a Future for Web-based Learning Objects*. Proceedings of International Conference on Web Engineering, Oviedo, Asturias, Spain, 2003.
- [11] Moodle MOODLE. Modular Oriented-Object Dynamic Learning Environment. Disponível em: <<http://www.moodle.org>>. Acesso em: 28 jun. 2007.
- [12] Noy, N. F.; Fergenson, R. W.; Musen, M. A. *The knowledge model of Protégé-2000: combining interoperability and flexibility*. In: Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management. ISBN: 3-540-41119-4, 2000.
- [13] Silveira, R. A., Gomes, E. R, Vicari, R. M. *Intelligent Learning Objects: An Agent-Based Approach of Learning Objects*. In Weert, Tom Van, Tatnall, Arthur (Eds.) Information and Communication Technologies and Real-Life Learning. Boston Springer, 1103 – 110, 2005.
- [14] Sosteric, M.; Hesmeier, S. *When is a Learning Object not an Object: A first step towards a theory of learning objects*. International Review of Research in Open and Distance Learning. ISSN: 1492-3831, 2002.