
Proposta de um Sistema Multi-agentes para a aplicação de Objetos Inteligentes de Aprendizagem seguindo o padrão SCORM

Júlia M. C. da Silva^{1,2}, Natanael Bavaresco¹, Ricardo Azambuja Silveira¹

¹Departamento Informática e Estatística – Universidade Federal de Santa Catarina
88040-900 – Florianópolis, SC – Brasil

²Centro de Ciências Tecnológicas da Terra e do Mar – Universidade do Vale do Itajaí
São José, SC – Brasil

{julia, natanael, silveira}@inf.ufsc.br

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 Multiagent 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) methodology.*

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).*

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 Silveira *et al.* (2005), 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 Costa (1999), 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 (Downes, 2001; Mohan e Brooks, 2003; e Sosteric e Hesemeier, 2002). Os objetos podem ser controlados por algum Sistema Gerenciador de Aprendizagem (*Learning Management Systems - LMS*), contudo estão limitados a funcionar de uma forma restrita.

Silveira *et al.* (2005), 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 (Bradshaw, 1997):

- Um agente é um pedaço de software que trabalha de forma contínua e autônoma, dentro de um ambiente;
- Um agente pode inferir 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.

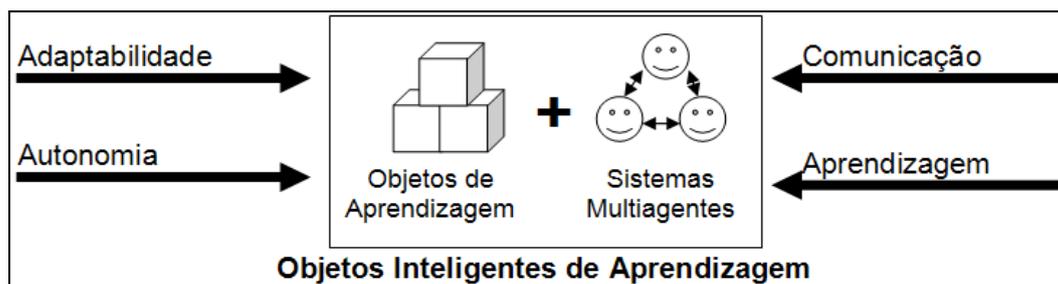


Figura 1. Objetos Inteligentes de Aprendizagem.

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 (ADL, 2001):

- **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* (ADL, 2001).

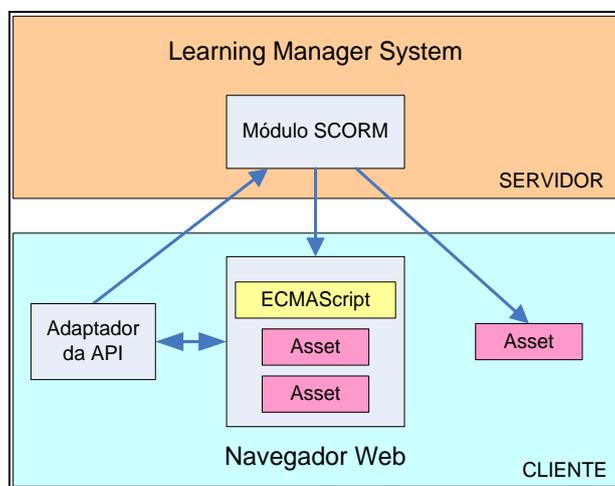


Figura 2. Ambiente de Run-Time

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 conteudista. 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 um 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.

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	Fornecer informações adicionais sobre um determinado erro corrente.

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”.

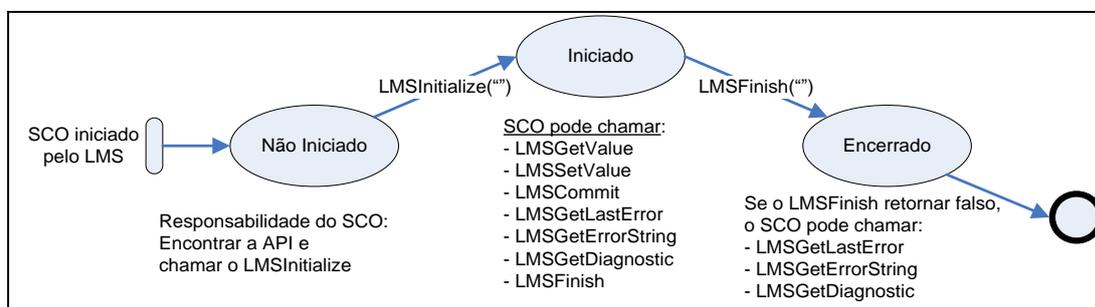


Figura 3. Estados de transições da API

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 contemplasse 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.

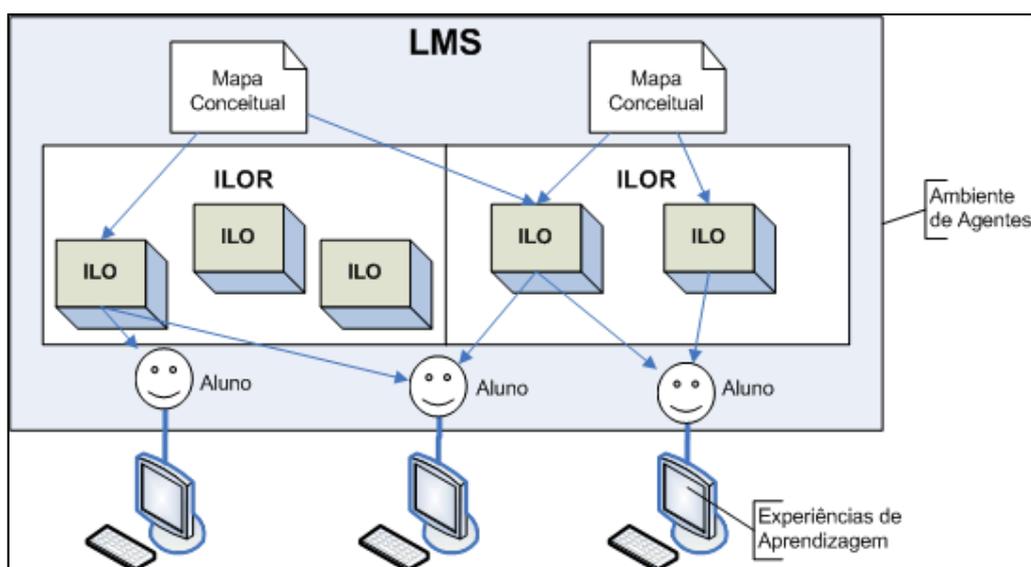


Figura 4. Sistema multi-agentes proposto por Silveira *et al.* (2005)

5. Modelando o Sistema Multi-Agente

O SMA proposto por Silveira *et al.* (2005) foi revisado e modelado através da metodologia proposta por DeLoach e Wood (2001), denominada MaSE. Ela permite o desenvolvimento de sistemas multi-agentes 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.

Conforme descrito na Tabela 1, estabeleceram-se cinco classes de agentes que representarão os papéis de agente LMS, agente ILOR e agente ILO, agente Learner, agente APISCORM; no que diz respeito a retornar uma determinada informação solicitada. O agente APISCORM é classificado como reativo, enquanto os demais são considerados agentes cognitivos.

Tabela 1. Agentes e diálogos da ontologia ilo-ontology.

Receptor	Diálogo	Protocolo FIPA	Descrição
APISCORMAgent	lms-initialize	RIPS	Informa ao SMA que o acesso ao objeto foi iniciado.
	lms-finalize	RIPS	Informa ao SMA que o acesso ao objeto foi finalizado.
LMSAgent	new-learner	RIPS	Registra um novo aluno.
	set-ilor	RIPS	Registra um novo repositório.
LearnerAgent	set-learner	RIPS	Solicita o armazenamento das informações de um aluno.
	get-learner	RIPS	Obtém informações gerais sobre um aluno.
ILORAgent	search-keyword	CNIPS	Obtém informações sobre ILOs que atendam a um determinado critério.
	new-ilo	RIPS	Registra um novo objeto inteligente de aprendizagem.
ILOAgent	get-metadata	RIPS	Obtém informações de metadados de um ILO.
	show-ilo	RIPS	Apresenta o ILO ao aluno.
	search-ilo	CNIPS	Busca um ILO que atendam a um determinado critério.

Legenda:

RIPS – Request Interaction Protocol Specification (SC00026)

CNIPS –Contract Net Interaction Protocol Specification (SC00029)

Já o diagrama de diálogos permite conhecer as mensagens que serão trocadas entre duas classes de agentes. Para os agentes identificados foram estabelecidos doze diálogos, os quais compõem a ontologia **ilo-ontology**, desenvolvida especialmente para a aplicação no domínio de Objetos Inteligentes de Aprendizagem.

Os diálogos da ontologia **ilo-ontology** seguem os protocolos de interação da especificação FIPA. Por exemplo, o fluxo de mensagens para o diálogo “*get-metadata*”, ilustrado na Figura 5, baseia-se no documento SC00026 – FIPA *Request Interaction Protocol Specification*.

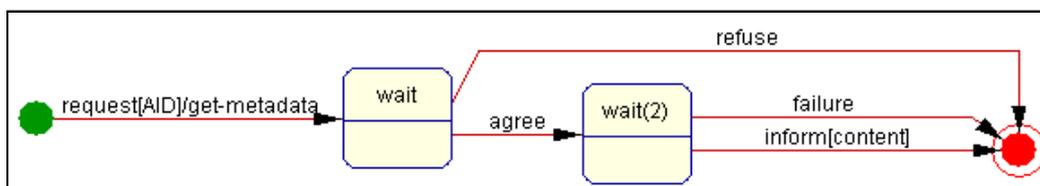


Figura 5. Diagrama do diálogo *get-metadata*.

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.

5.1. Ontologia ILO-Ontology

Conforme a FIPA (2002), 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 (Noy *et al.*, 2000), 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 6. 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.

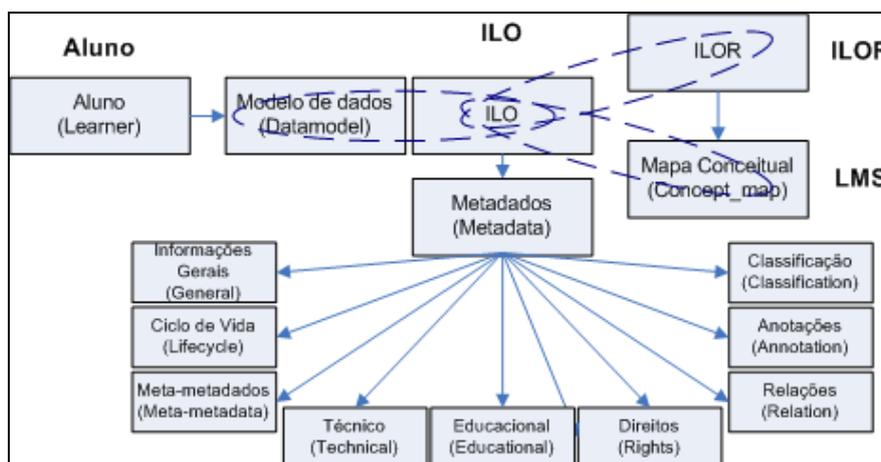


Figura 6. Conceitos da Ontologia.

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), 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. Aplicação

Foram realizadas experimentações de desenvolvimento de objetos de aprendizagem através de uma ferramenta de autoria. Selecionou-se a ferramenta Click2Learn (Click2Learn, 2007) por trabalhar com metadados SCORM, além de ser disponibilizada gratuitamente. Seu funcionamento consiste na seleção de um documento já construído previamente (ex: texto, imagem, hipertexto), o qual receberá meta-informações preenchidas pelo usuário através de um formulário. Ao final, é gerado um arquivo compactado contendo o objeto de aprendizagem “scormizado”. Este arquivo é formado por: um conjunto de outros arquivos que viabilizam a comunicação de um objeto com algum LMS, desde que trabalhem sob um mesmo padrão; o arquivo original que contém o material instrucional; e um arquivo de metadados disponibilizado no formato XML.

Este último arquivo é responsável por expressar as informações do objeto de aprendizagem, que serão carregados pelo agente ILO no diálogo “new-ilo”, executado no momento do registro de um novo objeto no repositório. Quando este diálogo ocorre, o agente prepara os dados do arquivo XML e disponibiliza seguindo a especificação da ontologia, conforme ilustra a Figura 7.

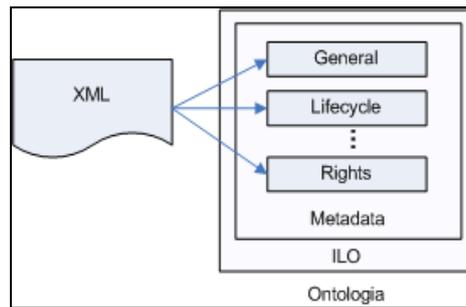


Figura 7. Conversão dos dados XML para a ontologia.

Através deste processo, torna-se viável a comunicação de um agente ILO com os demais agentes do sistema, por exemplo, podendo estabelecer o diálogo “*get-metadata*”.

7. Implementação

Para o desenvolvimento dos agentes está sendo utilizado o *framework* Java Agent Development - JADE (2007), na versão 3.4.1. O JADE implementa um conjunto de funcionalidades que facilita a implementação de agentes, além de disponibilizar ferramentas gráficas que apóiam o gerenciamento de SMAs. Os agentes gerados neste *framework* encontram-se no padrão FIPA.

O JADE não permite a instanciação nem a comunicação direta de seus agentes e serviços web, sejam aplicações cliente (*applet*) ou servidor (*jsp* ou *servlet*). Para tanto, foi necessário desenvolver um mecanismo de comunicação através de invocação remota de método (*Remote Method Invocation* - RMI), a qual especifica que se deve haver um servidor de serviços e clientes que irão se conectar a este servidor em busca de um determinado serviço. Logo, deve-se utilizar uma porta comum onde o serviço, também de nome conhecido, poderá ser localizado.

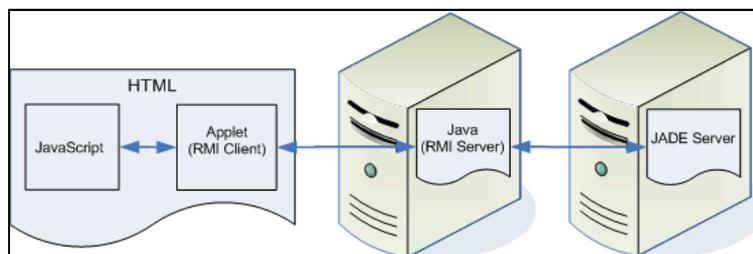


Figura 8. Integração Applet e SMA via RMI

O servidor deve realizar a comunicação com o SMA através dos recursos de *Gateway* disponibilizados pelo *framework* do JADE; e o cliente, desenvolvido com recursos de applet, conecta-se ao servidor. O serviço de *Gateway* permite a criação de um agente (*GatewayAgent*) que irá interagir com os demais agentes do SMA através da troca de mensagens, processo interpretado através da especificação de um comportamento.

8. Conclusões

Os Objetos Inteligentes de Aprendizagem são entidades capazes de promover experiências de aprendizagem através de ambientes de aprendizagem, utilizando como

base tecnológica o conceito de objetos de aprendizagem e sistemas multi-agentes. Isto permite a eles interagir com outros agentes, buscando conhecimento sobre o ambiente, alunos e demais objetos que possam auxiliá-lo no ensino de um domínio.

A sociedade proposta por Silveira *et al.* (2005), foi modelada 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. Por exemplo, anteriormente cabia ao agente de LMS solicitar a experiência de aprendizagem, o qual foi substituído pela ação de `lms-initialize` disparada pelo agente de `APIAdapter`. Outro agente modelado é o `LearnerAgent`, responsável pelas atividades de propriedade do aluno.

Ainda sobre a modelagem, a comunicação entre agentes foi revista, e propostos novos diálogos, a destacar aqueles que envolve a `APIAdapter`. Além disto, percebeu-se que os diálogos definidos utilizam apenas um processo definido de fases: requisição, confirmação, resposta ou rejeição. Cada um dos diálogos foram estudados quanto a sua forma de execução e definiu-se um protocolo de comunicação FIPA.

A implementação já realizada permite verificar a viabilidade do projeto, o qual se destaca a independência de servidor web para a sua implantação, necessitando apenas da plataforma JADE, com o SMA, e da adaptação da API de comunicação entre os objetos e o SMA.

Como trabalho futuro, além da continuidade da implementação e testes do SMA, está sendo reavaliada a ontologia proposta por Silveira *et al.* (2005), buscando conformidade com o conjunto de metadados do padrão SCORM.

Referências

- ADL (2001). The SCORM Run-Time Environment. Advanced Distributed Learning, 2001.
- Bradshaw, J. M. (1997). An introduction to software agents In: Bradshaw, J. M. Ed. Software Agents. Massachusetts: MIT Press, 1997.
- Click2Learn (2007). Disponível em: http://academiaelearning.com/contenido/scorm/cooking/i_cookbook.htm. Acessado em: 20 de abr. de 2007.
- Costa, M. T. C. (1999). Uma Arquitetura Baseada em Agentes para Suporte ao Ensino a Distância. Doutorado em Engenharia da Produção. Universidade Federal de Santa Catarina.
- DeLoach, S. A.; Wood, M. (2001). Developing Multiagent Systems with agentTool. In: Proceedings of Lecture Notes in Artificial Intelligence. Springer – Verlag. Berling, 2001.
- Downes, S. (2001). Learning objects: resources for distance education worldwide. In: International Review of Research in Open and Distance Learning, 2(1). 2001.

-
- FIPA: The foundation for Intelligent Physical Agents: Specifications (2002). Acessado em: 7 de Julho de 2005. Disponível em: <http://www.fipa.org>.
- 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.
- JADE (2007). Java Agent Development Framework. Acessado em: 25 de Junho de 2007. Disponível em: <http://jade.tilab.com>.
- Mohan, P.; Brooks, C. (2003). Engineering a Future for Web-based Learning Objects. Proceedings of International Conference on Web Engineering, Oviedo, Asturias, Spain.
- Noy, N. F.; Fergenson, R. W.; Musen, M. A. (2000) 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.
- Silveira, R. A., Gomes, E. R, Vicari, R. M. (2005). 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.
- Sosteric, M.; Hesmeier, S. (2002). 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.