Um Mecanismo para a Descoberta Automática de Ambientes Educacionais

Lucas Monteiro Braz
UNISINOS - Av. Unisinos, 950 - B. Cristo Rei / CEP 93.022-000 - São Leopoldo (RS)
GrOW – Grupo de Otimização na Web
lmonteirobraz@gmail.com

Ig Ibert Bittencourt
UFAL - Av. Lourival Melo Mota, s/n, Tabuleiro do Martins / CEP 57072-970 – Maceió (AL)
GrOW – Grupo de Otimização na Web
ig.ibert@ic.ufal.br

Evandro de Barros Costa
UFAL - Av. Lourival Melo Mota, s/n, Tabuleiro do Martins / CEP 57072-970 – Maceió (AL)
GrOW – Grupo de Otimização na Web
ebcosta@gmail.com

Resumo *Ambientes educacionais comumente falham em prover certos conteúdos, recursos educacionais e funcionalidades aos seus usuários devido ao grande número de estudantes e à diversidade de preferências e níveis de conhecimento entre eles. Essa situação motiva a interoperabilidade entre os ambientes educacionais, ou seja, a comunicação entre eles para o compartilhamento de objetos de aprendizagem. Entretanto, é necessário garantir que a interoperabilidade ocorra de maneira automática e escalável. Para tal, os ambientes devem ser capazes de descobrir automaticamente outros sistemas disponíveis para interoperar. Este trabalho apresenta um mecanismo desenvolvido para realizar a descoberta automática de ambientes educacionais e permitir que a interoperabilidade entre eles ocorra em larga escala.*

Palavras-Chave: *Interoperabilidade, Ambientes Educacionais, Ontologias*

Abstract *Educational environments often fail to provide certain content, educational resources and features to its users due to the large number of students and different preferences and levels of knowledge among them. This situation motivates the interoperability between educational environments, that is, the communication between them for the sharing of learning objects. However, it is necessary to ensure that interoperability occurs automatically and scalable. To this end, the environments must be able to automatically discover other available systems. This paper presents a mechanism to perform automatic discovery of educational environments in order to allow that interoperability occur on a large scale.*

Keywords: *Interoperability, Educational Environments, Ontologies*

Recebido: 21 de Abril de 2011 / Aceito: 08 de Agosto de 2011 / Publicado: 19 de Agosto de 2011
DOI: 10.5753/RBIE.2011.19.01.44

1 Introdução

Ambientes educacionais baseados na Web deveriam ser desenvolvidos de forma a suprir as necessidades cognitivas de todos os seus alunos. No entanto, diante da quantidade de estudantes utilizando esse tipo de sistema, cada um deles com diferentes níveis de conhecimento, objetivos, interesses e preferências na forma de aprendizado, a tarefa de construir um ambiente educacional completo o suficiente para atender a essas necessidades se torna bastante difícil. Portanto, é comum esses ambientes falharem ao prover certos recursos, conteúdos e funcionalidades aos seus estudantes. Esse fato pode acarretar em falhas no processo de ensino/aprendizado, insatisfação e desinteresse por parte dos estudantes, e até mesmo a desistência dos cursos.

Contudo, é possível que os objetos de aprendizagem inexistentes em um ambiente educacional, já tenham sido construídos no processo de autoria de outros ambientes. Então, não seria necessário refazer esses recursos, bastaria que os sistemas se comunicassem e compartilhassem os objetos de aprendizagem. De fato, os dois ambientes poderiam se beneficiar dessa interação.

A interoperabilidade entre ambientes educacionais possibilita a interação e colaboração entre esses sistemas para o compartilhamento de conteúdos, recursos educacionais, informações dos usuários e funcionalidades. Dessa forma, os ambientes passam a dispor de um conjunto mais rico de objetos de aprendizagem e tem mais opções para suprir as necessidades dos alunos, aumentando a sua eficácia e a satisfação dos estudantes.

Diversos trabalhos que tratam da interoperabilidade entre ambientes educacionais tem sido propostos pela comunidade científica (vide Seção 5). Em sua maioria, esses trabalhos garantem a interoperabilidade apenas de forma manual, ou seja, os desenvolvedores desses sistemas precisam codificar explicitamente as informações dos outros ambientes para que a interação possa ocorrer. Assim, quando é necessário interoperar com um novo ambiente, como mostrado na Figura 1, os desenvolvedores precisam especificar, implementar e configurar essa nova interação. Essa abordagem pode funcionar para um número reduzido de ambientes educacionais, mas à medida que esse número cresce, tal abordagem se torna inviável. Portanto, é necessário algum mecanismo para fazer com que essas interações possam ser definidas dinamicamente pelos sistemas (e não pelos desenvolvedores), permitindo que a interoperabilidade ocorra em larga escala. Ou seja, os ambientes precisam ser capazes de descobrir automaticamente outros ambientes disponíveis para interoperar. De fato, este é um dos principais problemas que concernem a interoperabilidade entre ambientes educacionais.

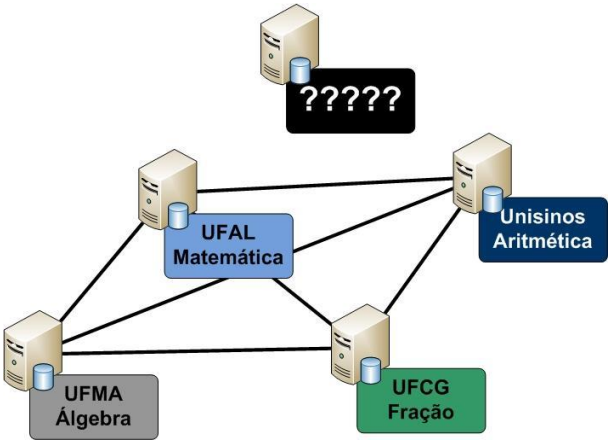

Figura 1 : Interoperabilidade com um novo ambiente. (Fonte [1])

O objetivo deste trabalho foi o desenvolvimento de um mecanismo para a descoberta de ambientes educacionais de maneira automática e escalável. O mecanismo proposto permite aos ambientes definir dinamicamente as interações com outros sistemas previamente desconhecidos, diferentemente das abordagens atuais, onde os programadores precisam configurar e implementar cada uma das interações e, para isso, devem conhecer os ambientes a priori.

Este artigo está estruturado da seguinte forma. Na Seção 2, são apresentados os conceitos relacionados à interoperabilidade de ambientes educacionais. A Seção 3 apresenta o mecanismo proposto, sua arquitetura e detalhes de implementação. Na Seção 4, é apresentado um estudo de caso que demonstra como ocorre a interoperabilidade utilizando o mecanismo proposto. Os trabalhos relacionados são mostrados na Seção 5 e na Seção 6 são feitas considerações finais deste trabalho.

2 Interoperabilidade entre Ambientes Educacionais

A interoperabilidade entre ambientes educacionais consiste na comunicação entre dois ou mais ambientes com o objetivo de que eles colaborem uns com os outros, compartilhando conteúdos, recursos educacionais, modelos dos usuários e até mesmo funcionalidades. Dessa forma, os ambientes ficam mais completos, mais eficazes, pois possuem um conjunto mais rico de recursos disponíveis aos seus usuários [2]. Além disso, o tempo de desenvolvimento desses ambientes é reduzido, visto que é possível reutilizar recursos previamente construídos.

Diversos motivos podem levar um sistema educacional a interagir com outros ambientes. Os principais são listados a seguir:

(i) Objetos de Aprendizagem: um dos principais motivos para a interoperabilidade é o compartilhamento de objetos de aprendizagem, como conteúdos, exemplos, problemas, exercícios;

(ii) Funcionalidades: caso um ambiente necessite de alguma funcionalidade que ele não é capaz de realizar, outro ambiente pode lhe prover tal funcionalidade;

(iii) Recomendações: alguns estudantes podem precisar de recomendações de pares para auxiliar no processo de aprendizagem. Se não houver pares disponíveis ou adequados, o sistema pode recomendar um par de outro ambiente, os quais podem ser estudantes, professores ou agentes tutores;

(iv) Modelos dos usuários: Ambientes Educacionais Adaptativos (AEH) coletam informações sobre seus usuários e as utilizam para adaptar seu comportamento de acordo com cada estudante [3]. Se um aluno utiliza (ou já utilizou) mais de um ambiente, estes podem compartilhar as informações desse estudante para construir modelos mais completos, fazendo com que os resultados das adaptações sejam mais adequados [4].

Para um melhor entendimento do conteúdo desta Seção e do restante do trabalho, introduziremos o conceito de serviço de interoperabilidade: um serviço disponibilizado por um ambiente educacional para que outros sistemas possam interagir com ele. Assim, quando um ambiente **A** solicita um recurso educacional do tipo problema a um ambiente **B**, consideramos que **A** está "invocando" um dos serviços de interoperabilidade do ambiente **B**. Nesse caso o serviço para o compartilhamento de problemas.

2.1 Como a Interoperabilidade Ocorre

Basicamente, os passos necessários para que a interoperabilidade ocorra são:

- Passo 1 - Identificação: o primeiro passo é identificar os momentos onde a interoperabilidade é necessária. Normalmente esses momentos ocorrem em uma das situações descritas anteriormente, como a necessidade de um objeto de aprendizagem;
- Passo 2 - Descoberta: após a identificação de que a interoperabilidade é necessária, é preciso descobrir com qual ambiente interagir. Além disso, é preciso determinar qual, dentre todos os seus serviços de interoperabilidade, deve ser "invocado";
- Passo 3 - Comunicação: quando o outro ambiente e seu serviço de interoperabilidade são descobertos, os dois sistemas podem se comunicar. Entretanto, é necessário garantir a comunicação entre esses sistemas levando-se em consideração suas tecnologias, como plataforma, paradigma de programação, entre outros;
- Passo 4 - Negociação: para evitar problemas na comunicação (*e.g.* um ambiente não entender o que lhe é requisitado), os ambientes precisam negociar qual o formato dos dados a serem compartilhados e tornar claro qual recurso está sendo solicitado. Contudo, se os ambientes possuem a mesma infraestrutura e a mesma representação do conhecimento, este passo é desnecessário.

2.2 Problemas Relacionados com a Interoperabilidade

Diversos problemas decorrem da interoperabilidade entre ambientes educacionais. Alguns deles estão relacionados com o fato de os ambientes terem sido construídos com diferentes infraestruturas enquanto outros problemas são comuns a qualquer tipo de interoperabilidade.

Entre os problemas comuns à interoperabilidade, destacamos:

- Descoberta dos ambientes: visto que existem diversos ambientes educacionais espalhados pela Web, um dos principais problemas que surge é: como um sistema pode encontrar outro ambiente disponível para interoperar? Além disso, o fato de um ambiente estar disponível não implica que seja interessante interagir com ele. Então, é preciso realizar um *"matching"* entre os sistemas, ou seja, comparar as suas características para verificar se o ambiente descoberto está apto a suprir as necessidades do ambiente que necessita interoperar;
- Descoberta do serviço de interoperabilidade: assim como o problema anterior, cada ambiente educacional pode prover diversos serviços de interoperabilidade. Então é necessário determinar qual serviço desse ambiente é o mais apropriado;
- Conceitos: a comunicação entre os ambientes ocorre baseada em conceitos. Por exemplo, um ambiente pode solicitar problemas do conceito "mul-

tiplicação" ou requisitar um especialista no conceito "fração". Dessa forma pode existir problemas relacionados com o uso de: linguagens diferentes, sinônimos ou palavras escritas da mesma maneira mas que representam conceitos distintos.

Quando os ambientes possuem infraestruturas diferentes, a interoperabilidade se torna mais complicada. Além dos problemas já citados é necessário lidar também com:

- Diferentes formatos de dados: um ambiente educacional deve ser capaz de interpretar os dados compartilhados, os quais possivelmente estarão em um formato diferente do que esse ambiente utiliza. Caso contrário a interoperabilidade não obterá sucesso;
- Tecnologias: é possível que os ambientes utilizem tecnologias diferentes, como linguagens ou paradigmas de programação distintos. Por exemplo, pode ser necessário comunicar um ambiente desenvolvido baseado em agentes com outro orientado a componentes.

O objetivo deste trabalho é oferecer uma solução para resolver os problemas de descoberta de ambientes educacionais e seus serviços de interoperabilidade.

3 Mecanismo Proposto

O mecanismo desenvolvido permite a ambientes educacionais realizarem a descoberta automática de outros ambientes disponíveis para interoperar e que tenham sido construídos com a mesma infraestrutura.

A arquitetura do mecanismo proposto consiste de três componentes fundamentais: ***i) ontologia de aplicação***, que armazena as informações, do ambiente educacional, relevantes para a interoperabilidade; ***ii) serviço de registro***: um serviço web utilizado para registrar os ambientes educacionais; e ***iii) serviço de descoberta***: serviço web responsável por realizar a descoberta dos ambientes educacionais. Ambos os serviços são disponibilizados por meio do *servidor de interoperabilidade*, onde também ficam armazenadas as informações dos ambientes registrados.

Para realizar a interoperabilidade, um ambiente educacional deve seguir os passos abaixo:

- **Registro**: o primeiro passo é invocar o *serviço de registro* para registrar o ambiente no *servidor de interoperabilidade*. Com isso, o ambiente fica apto a descobrir e ser descoberto, pois apenas ambientes registrados podem interoperar;
- **Descoberta**: quando a interoperabilidade é necessária, o ambiente deve descobrir outro sistema disponível capaz de suprir as suas necessidades. A descoberta é realizada através da invocação do *serviço de descoberta*, o qual retornará as informações necessárias para se comunicar com o ambiente descoberto;
- **Comunicação**: os ambientes devem se comunicar para compartilhar os recursos desejados.

É importante frisar que todo ambiente que deseja interoperar precisa obrigatoriamente estar registrado no *servidor de interoperabilidade*. Ao efetuarem o registro os ambientes deixam claro sua disponibilidade para cooperação. Assim, o processo de descoberta consiste basicamente em realizar uma busca nesse servidor e retornar um dos ambientes registrados. No entanto, essa abordagem pode retornar ambientes indesejados como, por exemplo, retornar um ambiente no domínio de literatura ao se buscar por exercícios de multiplicação. Para resolver esse problema, cada ambiente deve prover informações sobre as suas características (*e.g.* domínio do ambiente), de forma que seja possível compará-las e determinar o ambiente com a maior probabilidade de suprir as necessidades do ambiente solicitante.

Para garantir a automatização do processo de descoberta, as informações dos ambientes devem ser expressas de maneira estruturada e interpretável por máquinas, permitindo a construção de entidades de software que compreendam essas dados para realizar a comparação entre os ambientes. Para tal, essas informações devem ser providas por meio de uma ontologia. Nesse caso, *ontologia de aplicação* (vide Seção 3.1). Portanto, para realizar o registro, todo ambiente deve disponibilizar uma instância da *ontologia de aplicação* descrevendo suas características.

Para tornar mais claro como a interoperabilidade ocorre, suponhamos a existência de três ambientes educacionais: um ambiente de matemática, outro de medicina, e um terceiro para o ensino do sistema cardiovascular humano. Inicialmente, os ambientes precisam estar registrados, ou seja, cada um deles deve invocar o *serviço de registro*. Então, suponhamos que um estudante do ambiente de medicina esteja interessado em aprender os detalhes de como ocorre o transporte de nutrientes no corpo humano (função desempenhada pelo sistema cardiovascular). Como esse não é o foco do ambiente de medicina, ele aborda o assunto apenas de maneira superficial. Logo, para suprir as necessidades do estudante, ele decide interoperar. Para tal, o ambiente de medicina invoca o *serviço de descoberta*, o qual verifica a existência de dois ambientes (o de matemática e o de sistema cardiovascular), compara as ontologias dos ambientes registrados

com a do ambiente de medicina e determina que o ambiente mais adequado é o de sistema cardiovascular. Como resultado, o ambiente de medicina recebe as informações do ambiente descoberto necessárias para que a comunicação possa ser realizada. Finalmente, os ambientes se comunicam, trocam o recurso desejado, e o ambiente de medicina pode apresentar ao usuário o conteúdo que ele necessita.

Nesse exemplo verifica-se: *i)* a capacidade de um ambiente educacional descobrir outros sistemas de maneira simples e automática, *ii)* que o mecanismo de descoberta evita retornar um ambiente inadequado (nesse caso o ambiente de matemática), e *iii)* que todo o processo de interoperabilidade ocorre de maneira transparente ao usuário final do sistema.

Na Figura 2 é apresentada a arquitetura do mecanismo proposto e seus componentes, a qual representa uma extensão da arquitetura genérica proposta em [5]. Nas subseções seguintes, cada um dos componentes é descrito em detalhes.

3.1 Ontologia de Aplicação

A *ontologia de aplicação(ApplicationOntology.owl)* descreve semanticamente as características dos ambientes úteis e necessárias para realizar a tarefa de interoperabilidade. Essa ontologia é essencial, pois oferece suporte ao processo de comparação dos ambientes de forma automática e de definição do grau de similaridade entre eles. O uso dessa ontologia evita a recomendação de ambientes inadequados pelo *serviço de descoberta* (vide Seção 3.3). A Tabela 1 demonstra esta ontologia, a qual foi construída utilizando-se a linguagem OWL DL.

Environment $\sqsubseteq$ = id
Environment $\sqsubseteq$ = host
Environment $\sqsubseteq \geq$ 1 hasCategory
Environment $\sqsubseteq \geq$ 1 providesService Service
Service $\sqsubseteq \geq$ 1 hasCategory
Service $\sqsubseteq$ hasInput
Service $\sqsubseteq \leq$ 1 hasOutput
Entity $\sqsubseteq \geq$ 1 host
Entity $\sqsubseteq \geq$ 1 hasCategory
Entity $\sqsubseteq$ hasService

Tabela 1: Ontologia de Aplicação.

Em seguida descreve-se cada uma das classes dessa ontologia:

- **Environment:** define as informações gerais do ambiente educacional. Possui as seguintes propriedades: *id*, que deve armazenar um identificador recebido pelo ambiente após o registro; *host*, a qual armazena a URL do ambiente; *hasCategory*, por meio da qual é possível definir uma ou mais categorias para o ambiente; e *providesService*, que define os serviços de interoperabilidade deste ambiente;
- **Service:** representa um serviço de interoperabilidade provido pelo ambiente educacional. As pro-

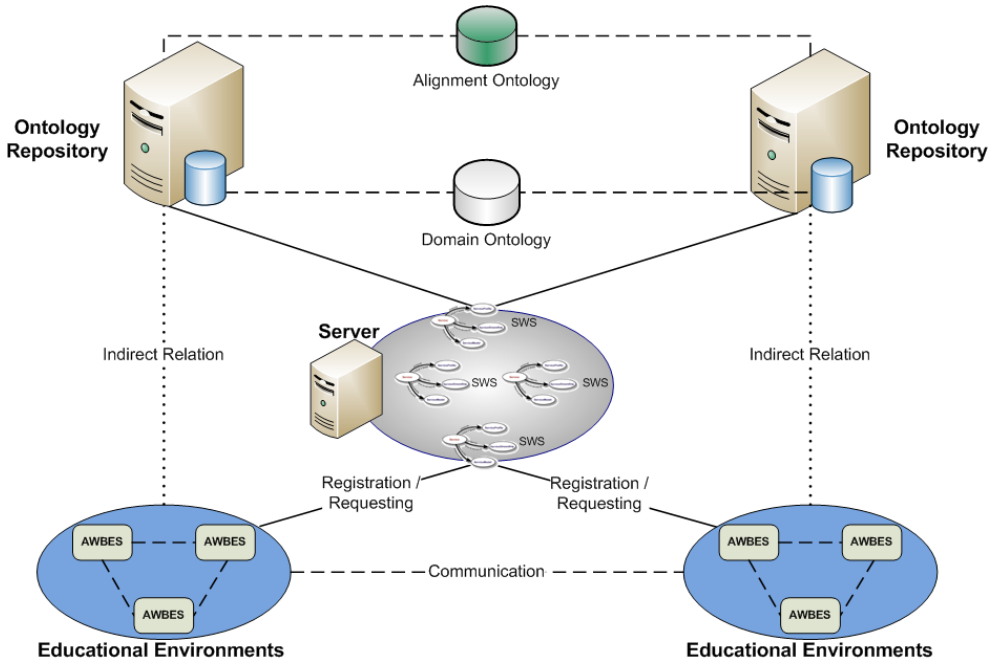

Figura 2: Arquitetura proposta. (Fonte [1])

priedades desta classe são: *hasCategory*, que define as categorias do serviço; *hasInput* e *hasOutput*, as quais representam, respectivamente, os parâmetros de entrada e o retorno do serviço;

- **Entity:** representa a entidade que de fato realiza um determinado serviço, por exemplo um agente ou componente. Suas propriedades são: *host*, que é uma referência para esta entidade; *hasService*, os serviços que esta entidade implementa; e *entityType*, para definir o tipo da entidade (*e.g.* agente, componente, classe).

As classes *Entity* e *Service* fornecem as informações necessárias para realizar a comunicação entre os ambientes educacionais. Por meio da propriedade *host*, a classe *Entity* informa com quem se deve comunicar, enquanto a classe *Service* informa "o que deve ser dito" através da propriedade *hasInput*.

A propriedade *hasCategory* permite definir uma ou mais categorias para as classes *Environment*, *Service* e *Entity*. Uma categoria de uma classe representa o contexto relacionado a essa classe. Por exemplo, um ambiente poderia ter as categorias "Matemática" ou "Medicina" representando o domínio desse ambiente. O conjunto de valores que essa propriedade pode assumir é definido na ontologia *Classification.owl*. Ou seja, o valor da propriedade *hasCategory* deve ser a *URI* de uma das classes presentes em tal ontologia. A Figura 3 representa uma partição da ontologia *Classification.owl* onde é possível verificar que as categorias são organizadas de maneira hierárquica.

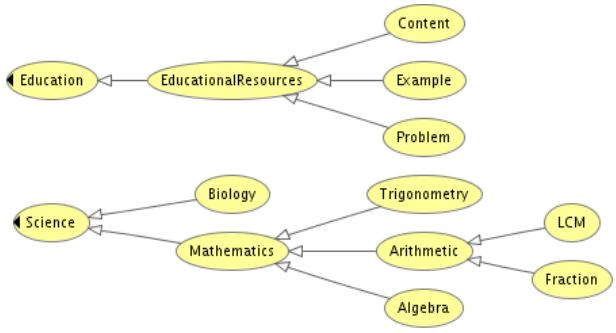

Figura 3: A ontologia de categorias, *Classification.owl*.

3.2 Serviço de Registro

Este serviço tem a função de registrar os ambientes educacionais no *servidor de interoperabilidade* e, consequentemente, permitir que eles possam descobrir e serem descobertos por outros ambientes. Para realizar o registro, um ambiente deve possuir, obrigatoriamente, uma instância da ontologia *ApplicationOntology.owl* disponível na Web. Em seguida, basta invocar este serviço passando os parâmetros adequados.

O *serviço de registro* armazena os serviços registrados no arquivo *EnvironmentRepository.xml*. Esse serviço recebe como parâmetro de entrada a *URI* da *ontologia de aplicação* do ambiente educacional e verifica se essa *URI* já está registrada no arquivo *EnvironmentRepository.xml*. Caso já esteja, uma exceção é lançada, caso contrário o serviço gera um identificador para o ambiente e o escreve no arquivo xml associado à *URI* do ambiente. Por fim, o serviço retorna o identificador gerado.

3.3 Serviço de Descoberta

O *serviço de descoberta* é responsável por realizar a descoberta automática de ambientes educacionais disponíveis para a interoperabilidade. Esse serviço calcula a similaridade entre o sistema que o invocou e cada um dos ambientes registrados, retornando as informações do ambiente mais similar. Assim, o sistema que o invocou pode se comunicar com o ambiente descoberto e solicitar os recursos que necessita.

Quando um ambiente educacional decide interoperar, ele necessita que algum outro ambiente realize uma função que ele não é capaz de fazer, ou seja, ele precisa "invocar" um serviço de interoperabilidade provido pelo outro ambiente. Como um ambiente educacional pode prover mais de um serviço de interoperabilidade, apenas a informação do ambiente não é suficiente. É necessário também saber qual serviço de interoperabilidade desse ambiente "invocar". Portanto, além de descobrir um ambiente educacional, o *serviço de descoberta* deve determinar também qual serviço desse ambiente é o mais adequado.

Ao invocar o *serviço de descoberta* o ambiente deve informar o seu identificador e as características do serviço de interoperabilidade que ele desejaria que lhe fosse provido. Dessa forma, o *serviço de descoberta* deve receber como parâmetro de entrada a tupla *<environmentId, serviceInput, serviceOutput, serviceCategory>* onde: *environmentId* é o identificador do ambiente que invocou o *serviço de descoberta*; *serviceInput* corresponde ao conjunto de entradas do serviço desejado; *serviceOutput* é a saída esperada desse serviço; e *serviceCategory* é o conjunto de categorias desse serviço (*e.g.* "Matemática", "Biologia").

O serviço realiza os seguintes passos: ***i)*** recupera todos os ambientes registrados; ***ii)*** para cada ambiente, recupera todos os serviços; ***iii)*** calcula a similaridade entre cada serviço e o serviço desejado (*i.e.* o serviço que o ambiente passou como parâmetro); ***iv)*** determina o serviço com a maior semelhança; ***v)*** calcula a similaridade entre o ambiente registrado e aquele que invocou o serviço; ***vi)*** determina o ambiente mais similar; ***vii)*** retor-

na as informações do ambiente descoberto (e do seu serviço).

Para realizar a comparação entre os ambientes o *serviço de interoperabilidade* faz uso de métricas de similaridade. Em geral, uma métrica de similaridade é utilizada para comparar duas entidades (neste caso, os ambientes) e determinar quão similares elas são. Inicialmente foram implementadas três métricas: a *Jaccard Similarity*, a métrica do cosseno e a da distância euclidiana. Por meio de um arquivo de configuração é possível determinar qual métrica será utilizada. Além disso, foi utilizado o padrão de projeto *Strategy* para possibilitar a implementação de novas métricas caso haja necessidade. As métricas implementadas são descritas em [6].

Dessa forma, sejam E_1 o ambiente que invocou o *serviço de descoberta* e E_2 um dos ambientes registrados no *servidor de interoperabilidade*. Para calcular a similaridade entre esses ambientes, inicialmente é preciso determinar qual serviço do ambiente E_2 deve ser "invocado". Assim, sejam S_D o serviço desejado por E_1; S um serviço qualquer de E_2; s_I, s_O e s_C os atributos de S_D *serviceInput*, *serviceOutput* e *serviceCategory*, respectivamente; e s'_I, s'_O e s'_C os mesmos atributos do serviço S. Consideremos ainda que e'_C seja o conjunto de categorias da entidade que provê o serviço S. A similaridade entre S_D e S é dada por:

$$Sim(S_D,S) = \frac{1\times\sigma(s_I,s'_I)}{7} + \frac{2\times\omega(s_O,s'_O)}{7} + \frac{2\times\sigma(s_C,s'_C)}{7} + \frac{1\times\sigma(s_C,e'_C)}{7} \quad \textbf{(1)}$$

onde ω(a,b) = 1, se a = b e ω(a,b) = 0 caso contrário; e σ(a,b) é o cálculo da similaridade entre a e b utilizando uma das métricas implementadas.

Para calcular a similaridade entre os ambientes E_1 e E_2, utiliza-se a Equação 1 para encontrar a similaridade entre o serviço desejado por E_1 e cada um dos serviços providos por E_2. Então, identifica-se o serviço com a maior similaridade e o valor dessa similaridade é atribuído à variável Δ na Equação 2 para calcular a similaridade entre E_1 e E_2

$$Sim(E_1,E_2) = \frac{1\times\sigma(E_1C,E_2C)}{7} + \Delta \quad \textbf{(2)}$$

onde E_1C e E_2C são os conjuntos de categorias dos ambientes E_1 e E_2, respectivamente, e o valor de $\sigma(E_1C, E_2C)$ é calculado como descrito anteriormente.

Para tornar claro como as métricas de similaridade são utilizadas, consideremos as categorias s_C = [*#LCM*] e s'_C = [*#LCM, #Example*] (vide Figura 3). Inicialmente, o algoritmo definido em [7] é utilizado para transformar essas categorias em vetores n-dimensionais, como descrito nos passos abaixo:

- Passo 1 - Definir vetores com os ancestrais de cada categoria e definir a união desses vetores:

*(LCM) = (LCM, Arithmetic, Mathematics, Science)

*(LCM/Example) = (LCM, Arithmetic, Mathematics, Science, Example, Educational Resources, Education)

*(LCM *U* LCM/Example) = (LCM, Arithmetic, Mathematics, Science, Example, Educational Resources, Education)

- Passo 2 - Construir um novo vetor em que: caso uma entidade definida no vetor união seja um dos ancestrais da categoria, adiciona-se 1 na posição correspondente do novo vetor, ou 0 caso contrário:

*Construir o vetor V_1:

(LCM *U* LCM/Example) = (LCM, Arithmetic, Mathematics, Science, Example, Educational Resources, Education)

(LCM) = (LCM, Arithmetic, Mathematics, Science)

V_1 = (1,1,1,1,0,0,0)

*Construir o vetor V_2:

(LCM *U* LCM/Example) = (LCM, Arithmetic, Mathematics, Science, Example, Educational Resources, Education)

(LCM/Example) = (LCM, Arithmetic, Mathematics, Science, Example, Educational Resources, Education)

V_2 = (1,1,1,1,1,1,1)

- Passo 3 - Por fim, utiliza-se uma das métricas para calcular a similaridade entre os vetores V_1 *e* V_2. Caso a métrica escolhida seja a métrica do cosseno temos:

$$SimCos(\vec{V_1},\vec{V_2}) = \frac{\vec{V_1}.\vec{V_2}}{\|\vec{V_1}\|.\|\vec{V_2}\|} = \frac{4}{2\times\sqrt{7}} = 0{,}75$$

A Seção seguinte apresenta um estudo de caso que torna mais claro como essas equações são utilizadas para calcular a similaridade entre dois ambientes educacionais.

4 Estudo de Caso

Esta seção descreve um estudo de caso para demonstrar a utilização do mecanismo proposto no processo de interoperabilidade entre dois ambientes educacionais, sendo um ambiente para o domínio de fração, chamado FraW, e outro para o domínio de MMC (Mínimo Múltiplo Comum). Ambos possuem a mesma infraestrutura,

pois foram construídos com o *framework* Massayo-F.

O Massayo-F é um *framework* para construir Ambientes Educacionais Adaptativos e Semânticos cujo principal objetivo é aumentar a eficiência e o reuso no desenvolvimento dessas aplicações, garantindo o desenvolvimento com baixo custo e tempo, com o mínimo de modificação de código [1]. Esse *framework* utiliza agentes de software, ontologias, Serviços Web Semânticos, entre outras tecnologias.

Entre os componentes arquiteturais do Massayo-F destacamos os seguintes agentes de software: Agente Tutor, responsável por acompanhar os estudantes no processo de aprendizagem; e Agente Mediador, que possui, entre outras, a função de mediar a comunicação com outros ambientes educacionais para permitir a interoperabilidade entre eles.

4.1 FraW

O FraW (Frações na Web) , desenvolvido por Sibaldo et al. [8], é um Sistema Tutor Inteligente para o domínio de Fração que objetiva ajudar alunos das séries iniciais do ensino fundamental. A aprendizagem baseia-se na resolução de problemas e na verificação do desempenho do estudante. Este sistema está habilitado a: resolver problemas propostos pelos alunos, explicando cada passo utilizado na solução, e a avaliar as respostas apresentadas pelos estudantes, oferecendo ajuda em eventuais impasses no processo de elaboração da solução. A ajuda está fundamentada na elaboração de dicas específicas para o tópico ao qual o aluno está com dificuldade, apresentando seu erro e posteriormente explicando como ele deveria resolver a questão em foco. A Figura 4 mostra a tela do FraW.

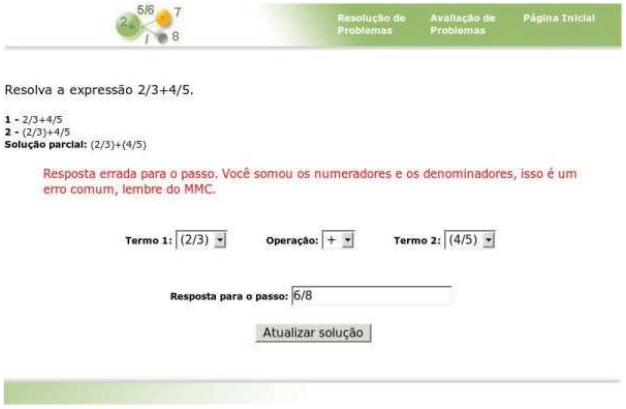

Figura 4: Tela do FraW. (Fonte [8])

Como descrito na Seção 3 cada ambiente educacional deve possuir uma instância da ontologia *ApplicationOntology.owl*. A Figura 5 representa a *ontologia de aplicação* do FraW.

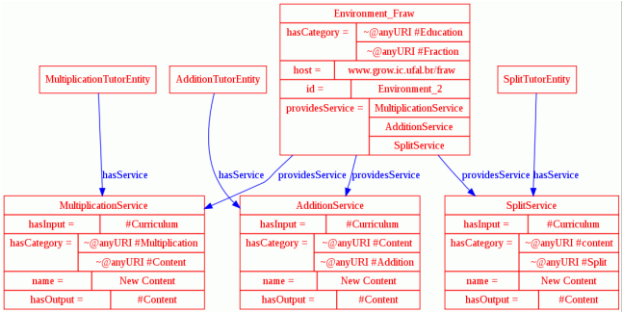

Figura 5: Ontologia de aplicação do FraW.

4.2 MMC

O ambiente educacional desenvolvido por Bittencourt [1] tem o objetivo de fazer com que os estudantes interajam para adquirir conhecimento no subdomínio da matemática de MMC (Mínimo Múltiplo Comum). A sua ideia é permitir a interação entre alunos e professores através de ferramentas como chat, fórum, enquete e wiki. A tela do ambiente de MMC é mostrada na Figura 6.

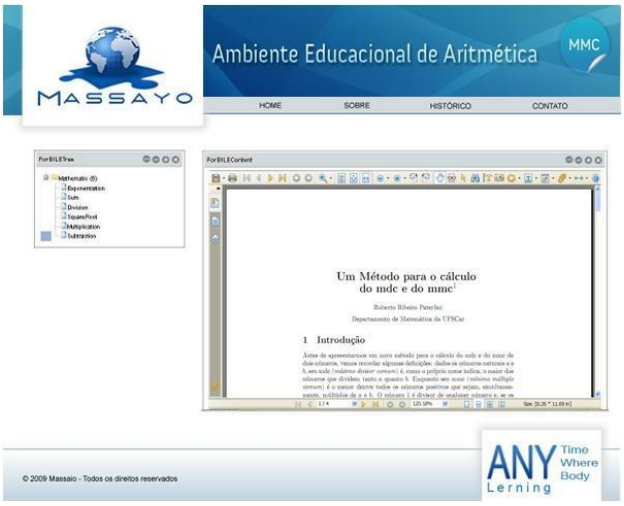

Figura 6: Tela do ambiente de MMC. (Fonte [1])

A partir da Figura 7, que representa a *ontologia de aplicação* deste ambiente, verifica-se que o ambiente de MMC provê os serviços de interoperabilidade para o compartilhamento de conteúdo, problemas e exemplos. Além disso, uma mesma entidade é responsável por todos os serviços.

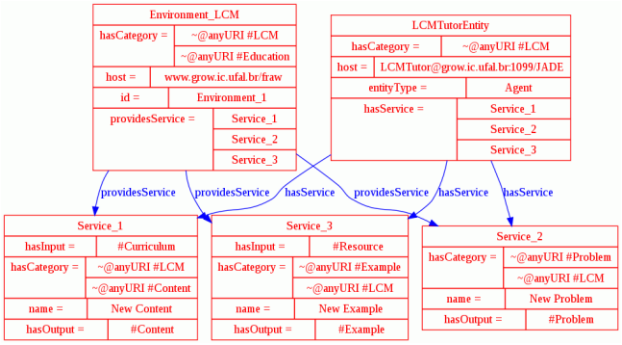

Figura 7: *Ontologia de aplicação* do ambiente de MMC.

4.3 Interação FraW - MMC

Neste cenário, um estudante do FraW (ambiente do domínio de fração) é submetido a uma série de problemas e acaba errando diversas questões. O agente tutor responsável por esse aluno analisa as suas respostas e verifica que os erros ocorreram em questões onde era necessário calcular o MMC. Logo, o agente chega a conclusão de que o aluno precisa de um reforço nesse assunto e decide que é necessário apresentar-lhe exemplos de como realizar o cálculo do MMC.

Como esse agente tutor não sabe "ensinar" o MMC, ele solicita ajuda ao agente mediador do ambiente, o qual verifica que o FraW não possui os objetos de aprendizagem requisitados (*i.e.* exemplos no cálculo do MMC) e decide que é necessário interoperar.

Em seguida são descritos os passos para que a interoperabilidade ocorra com sucesso.

Passo 1 - Registro: Para tornar a interoperabilidade possível, os ambientes devem estar registrados no *servidor de interoperabilidade*. A Figura 8 mostra o arquivo *EnvironmentRepository.xml* resultante dos registros dos ambientes, onde é possível verificar seus identificadores e *URIs*.

```
−<repository>
 −<environment>
  −<environmentURI>
     http://grow.ic.ufal.br/owl/ApplicationOntology.owl#Environment_LCM
   </environmentURI>
   <id>Environment_1</id>
 </environment>
 −<environment>
  −<environmentURI>
     http://grow.ic.ufal.br/owl/ApplicationOntology.owl#Environment_Fraction
   </environmentURI>
   <id>Environment_2</id>
 </environment>
</repository>
```

Figura 8: O arquivo *EnvironmentRepository.xml* após o registro dos ambientes.

Passo 2 - Descoberta: Uma vez que os ambientes estão registrados, o FraW pode descobrir outro ambiente para interagir. Neste cenário, o ambiente de fração deve invocar o *serviço de descoberta* passando como parâmetro o seu identificador (*Environment_2*) e o serviço desejado S_D, da forma:

serviceInput - S_ID = ["http://.../DomainCP.owl#Curriculum"]

serviceOutput - S_OD ="http://.../DomainResource.owl#Example"

serviceCategory - S_CD =["http://.../Classification.owl#LCM"]

Para calcular a similaridade entre o FraW e o ambiente de MMC, *o serviço de descoberta* deve computar a similaridade entre o serviço desejado e cada um dos serviços providos pelo ambiente de MMC. Para tal, utiliza-se a Equação 1 da seguinte forma[1]:

- Similaridade entre S_D e *Service_1*:

O serviço *Service_1* possui as seguintes propriedades:

serviceInput - S_I1 = ["http://.../DomainCP.owl#Curriculum", "http://.../Learner.owl#Learner"]

serviceOutput - S_O1 = "http://.../DomainResource.owl#Example"

serviceCategory - S_C1 = ["http://.../Classification.owl#LCM", "http://.../Classification.owl#Example"]

entityCategory - S_E1 = ["http://.../Classification.owl#LCM"]

Assim:

$$SimCos(S_D, Service\_1) = \frac{1\times 0{,}71}{7} + \frac{2\times 1}{7} + \frac{2\times 0{,}75}{7} + \frac{1\times 1}{7} = 0{,}74$$

- Similaridade entre S_D e *Service_2*:

O serviço *Service_2* é tal que:

serviceInput - S_I2 = NULL

serviceOutput - S_O2 = "http://.../DomainResource.owl#Problem"

serviceCategory - S_C2 = ["http://.../Classification.owl#Problem", "http://.../Classification.owl#LCM"]

entityCategory - S_E2 = ["http://.../Classification.owl#LCM"]

$$SimCos(S_D, Service\_2) = \frac{1\times 0}{7} + \frac{2\times 0}{7} + \frac{2\times 0{,}75}{7} + \frac{1\times 1}{7} = 0{,}36$$

- Similaridade entre S_D e *Service_3*:

O serviço *Service_3* possui as propriedades:

serviceInput - S_I3 = ["http://.../DomainCP.owl#Curriculum"]

serviceOutput - S_O3 = "http://.../Agent.owl#TutorAgent"

serviceCategory - S_C3 = ["http://.../Classification.owl#LCM"]

[1] Neste cenário foi utilizada a métrica de similaridade do cosseno.

entityCategory - S_E3 = ["http://.../Classification.owl#LCM"]

Logo:

$$SimCos(S_D, Service\_3) = \frac{1\times1}{7} + \frac{2\times0}{7} + \frac{2\times1}{7} + \frac{1\times1}{7} = 0{,}57$$

Assim, o serviço *Service_1* é o mais similar e, portanto, Δ assume o valor 0,74. Então, a similaridade entre os ambientes é:

Categorias do FraW = ["http://.../Classification.owl#Education", "http://.../Classification.owl#Fraction"]

Categorias MMC = ["http://.../Classification.owl#LCM", "http://.../Classification.owl#Education"]

$$SimCos(FraW, MMC) = \frac{1\times0{,}8}{7} + 0{,}74 = 0{,}84$$

Portanto, a similaridade entre os ambientes FraW e de MMC, utilizando-se a métrica do cosseno é 0,84.

Como resultado da execução do *serviço de descoberta*, o agente mediador do FraW recebe as informações do agente do ambiente de MMC que provê o serviço *Service_1* (pois esse foi o serviço que apresentou maior similaridade).

Passo 3 - Comunicação: De posse das informações do agente LCMTutor, o FraW pode realizar a comunicação e solicitar os recursos de que necessita. Este passo está além do escopo do mecanismo proposto, entretanto, visto que ambos os ambientes foram construídos com o *framework* Massayo-F, o qual possui suporte a este tipo de comunicação, não há necessidade de mecanismos adicionais para que a interoperabilidade ocorra.

Como resultado da interoperabilidade, o ambiente FraW pode satisfazer as necessidades de seu estudante, apresentando-lhe o conteúdo recebido do ambiente de MMC.

5 Trabalhos Relacionados

Diversos trabalhos tem abordado o problema da interoperabilidade entre ambientes educacionais. O objetivo desta seção é apresentar uma breve descrição de alguns trabalhos que demonstram alguma similaridade com este.

Em [9] é proposto um *framework* para realizar a interoperabilidade entre modelos de usuário através do uso de ontologias e de uma abordagem orientada a serviços. Esse *framework* apresenta técnicas de negociação baseada em diálogo que dão suporte à interações na forma de conversação e utiliza uma abordagem de registro dos ambientes em um servidor central. Contudo, esse *framework* delega para o ambiente a responsabilidade de realizar o cálculo da similaridade entre as aplicações, o que torna o desenvolvimento dos ambientes educacionais mais complexo, uma vez que os desenvolvedores precisam implementar os algoritmos de comparação para determinar com quem compartilhar informações.

Em [10], um modelo para garantir a interoperabilidade entre aplicações web semânticas é proposto. A solução faz uso de uma abordagem baseada no alinhamento de ontologias realizado através de POAF (*Portable Ontology Aligned Fragments*) e SWEDER (*Semantic Wrapping of Existing Data sources with Rules*). O aspecto positivo de tal abordagem é que POAF possui uma metodologia de alinhamento associada. No entanto, esse trabalho garante a interoperabilidade apenas a nível de ontologias, o que resulta em algumas dificuldades quando é necessário a adição de novos ambientes educacionais.

Brusilovsky [11] propõe uma abordagem para a integração de modelos dos estudantes coletados por diferentes sistemas educacionais. Esse trabalho utiliza uma abordagem baseada em protocolos para comunicação e no mapeamento manual dos modelos de domínio. Além disso, é apresentada a integração de dois ambientes educacionais, o QuizJET e o Java-Problets. Porém, a abordagem utilizada demanda que a inserção de um novo ambiente seja feita de maneira manual, ou seja, os desenvolvedores do sistema precisam explicitamente codificar as informações do novo ambiente.

As propostas descritas nesta Seção utilizam diferentes abordagens para realizar a interoperabilidade entre ambientes educacionais. Apesar de apresentarem resultados promissores, esses trabalhos falham em prover a interoperabilidade de maneira escalável e automática. Nessas propostas, para que a interoperabilidade ocorra, os desenvolvedores dos sistemas precisam conhecer a priori com quais ambientes irão interagir para especificar, implementar e configurar essas interações manualmente. Com isso, a interoperabilidade se torna mais complexa e demorada à medida que surgem novos ambientes com os quais se deseja interagir. Em contrapartida, o presente trabalho propõe um mecanismo para realizar a descoberta automática de ambientes educacionais, de forma que as interações são definidas dinamicamente, sem a necessidade da intervenção dos programadores do ambiente.

6 Considerações Finais

O principal objetivo deste trabalho foi apresentar o desenvolvimento de um mecanismo para realizar a descoberta automática de ambientes educacionais disponíveis para interoperabilidade. Tal mecanismo faz uso de ontologias para descrever as características de cada ambiente (como o domínio de aplicação e os serviços de interoperabilidade providos) de uma forma inteligível por

máquinas e utiliza métricas de similaridade para comparar as características descritas nas ontologias afim de determinar quão similares os ambientes são.

Para demonstrar a utilização do mecanismo desenvolvido, apresentou-se um estudo de caso onde ocorre a interoperabilidade entre dois ambientes educacionais, um para o domínio de fração e outro para o ensinos de Mínimo Múltiplo Comum.

A complexidade inerente ao problema acarreta em algumas limitações ao mecanismo desenvolvido. Ao realizar a descoberta dos ambientes educacionais, é possível que dois ambientes apresentem o mesmo grau de similaridade. Nessa caso, qualquer um será escolhido sem considerar a possibilidade de que um deles possa ser mais adequado que o outro. Portanto, verificações adicionais mais profundas precisariam ser realizadas. Além disso, o maior grau de similaridade encontrado poderia representar um valor geral baixo, digamos 10 por cento de similaridade. Assim, surge a necessidade de um limiar de corte. Contudo, são necessários mais dados a respeito do problema para realizar a escolha de um valor de corte adequado. Outro ponto a ser considerado é o fato de que as ontologias nas quais o mecanismo de comparação se baseia serão construídas manualmente pelos responsáveis de cada ambiente. Dessa forma, poderão ser introduzidos erros nas descrições dos ambientes e resultados indesejados serão obtidos.

Apesar das limitações, o mecanismo desenvolvido constitui-se em uma solução útil e funcional, como demonstrado no estudo de caso, para facilitar a comunicação e cooperação entre os ambientes, de forma que eles possam dispor de um conjunto mais rico de recursos educacionais.

Como principais trabalhos futuros, temos: *i)* extensão do mecanismo proposto para permitir a descoberta de ambientes educacionais quando estes possuem diferentes infraestruturas; *ii)* uma vez que é possível a utilização de várias métricas de similaridade, surge a necessidade de avalia-las para determinar qual a mais adequada para o problema em questão; *iii)* construir estudos de caso mais complexos, que envolvam um número maior de ambientes educacionais; *iv)* realizar comparações mais complexas entre ambientes com mesmo nível de similaridade; e *v)* avaliar e definir um limiar de corte;

Referências

[1] I. I. Bittencourt. Modelos e Ferramentas para a Construção de Sistemas Educacionais Adaptativos e Semânticos. Tese de Doutorado, Universidade Federal de Campina Grande, 2009.

[2] I. I. Bittencourt, E. Costa, L. M. Braz, H. Pacheco, D. Dicheva. Supporting Interoperability between Web-based Educational Systems. In *Proceedings of the 39th ASEE/IEEE Frontiers in Education Conference*, 2009.

[3] P. Brusilovsky. Adaptive Hypermedia. In *User Modeling and User-Adapted Interaction*, volume 11, páginas 87-110, 2001.

[4] J. Vassileva. Distributed User Modeling for Universal Information Access. *In HCI*, páginas 122-126, 2001.

[5] D. Dicheva, L. Aroyo. General Architecture Supporting Component-based EIS Interoperability. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), International Conference on Intelligent Tutoring Systems (ITS)*, 2004.

[6] C. D. Manning, P. Raghavan, H. Schutze. An Introduction to Information Retrieval. Cambridge University Press, 2009.

[7] H. Barros, I. Calado, M. Silva, I. I. Bittencourt, E. Costa. Using Semantic Web Services to Automatically Attend to Educational Requests. In *Anais do XX Simpósio Brasileiro de Informática na Educação*, 2009.

[8] M. A. Sibaldo, A. F. Rocha, F. M. Medeiros, I. I. Bittencourt, E. Costa. FraW - Ambiente Interativo de Aprendizagem para o Domínio de Fração via Web. In *Anais do XIX Simpósio Brasileiro de Informática na Educação*, 2008.

[9] F. Cena, F. Roberto. User Model Interoperability in a SOA Environment. In *AH*, páginas 284-297, 2008.

[10] D. Braines, Y. Kalfoglou, P. R. Smart, J. Bao, N. Shadbolt, J. Hendler. Semantic Web Techniques to Support Interoperability in Distributed Networked Environments. In *2nd Annual Conference of the International Technology Alliance (ACITA '08)*, 2008.

[11] P. Brusilovsky, S. Sosnovsky, M. Yudelson, A. Kumar, I. H. Hsiao. User Model Integration in a Distributed Adaptive E-Learning System. In *Proceedings of Workshop on User Model Integration at the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2008.