

Sistema Seleccionador de Recursos Computacionais de Grade baseado na Interoperabilidade de Múltiplas Ontologias

A.P.C. Silva, M.A.R. Dantas
Universidade de Santa Catarina (UFSC)
Departamento de Informática e Estatística (INE) - LaPeSD
88040-900 - Florianópolis - Brazil
{parra,mario}@inf.ufsc.br

Resumo

O paradigma de grade computacional tem como uma das suas principais características o compartilhamento de recursos heterogêneos espalhados geograficamente por diversas organizações virtuais. No entanto, o processo de seleção destes recursos torna-se na prática difícil, pois as organizações apresentam visões distintas quanto à forma de descrever seus recursos. Desta forma, em um sistema de seleção de recursos em grades é desejável realizar comparações semânticas. Neste artigo, propomos um sistema seleccionador de recursos que considera diferentes visões que as organizações podem apresentar. Este sistema apoia-se na abordagem de integração híbrida sobre regras semânticas para alcançar a interoperabilidade entre as diversas ontologias. Resultados experimentais práticos indicam que o sistema alcançou com sucesso o objetivo.

1. Introdução

A tecnologia de grade computacional é uma abordagem que propõe a redução de problemas relacionados ao compartilhamento e coordenação de recursos entre diferentes organizações, também conhecida como organizações virtuais (OVs) [10, 9].

Como mencionado em [7], um usuário (ou uma aplicação) deveria ser capaz de escolher tipos de recursos mais adequados dentro de uma configuração de grade de acordo com os requisitos de uma aplicação. O processo de seleção de recurso baseado em requisitos da aplicação é conhecido por *resource matching*.

A seleção de recursos está relacionada à linguagem usada para descrever as características dos recursos e aos requisitos da aplicação. Como observado em [23], o processo de seleção de recursos é muito limitado quando o processo de descrição e seleção é baseado no paradigma de

simetria de par de atributos. Esta abordagem considera que as descrições dos recursos são comparados sintaticamente aos requisitos descritos no pedido de recurso [23].

Considerando que OVs são entidades heterogêneas e independentes, é difícil de garantir a compatibilidade entre todas as descrições de recursos e de seus pedidos. Diferente de propostas convencionais, tais como mostrada em [5, 4], o trabalho de pesquisa publicado em [23] sugere uma abordagem mais flexível e extensível, usando o paradigma de ontologias para descrever os recursos para posteriormente serem selecionados.

O uso de ontologias para a descrição sintática e semântica de recursos computacionais [19] e realização de *matching* semântico [23, 1, 13, 15] tem aumentado nos últimos anos, porém, sem que contudo, exista uma ontologia única que represente os recursos de ambientes de grade [1, 15]. É mencionado em [11] que o ideal seria que houvesse uma ontologia padrão descrevendo cada modelo de diferentes domínios, ou seja, uma ontologia para cada área da medicina, uma para a área de negócios, etc. Todavia, isto não corresponde à realidade visto que existem múltiplas ontologias descrevendo os mesmos domínios, o que também vem sendo observado no domínio das grades.

Pesquisas [24, 3] têm utilizado ontologias como forma de expressar e integrar fontes de informações heterogêneas. A capacidade de interoperabilidade neste tipo de ambiente é interessante. Contudo, a interoperabilidade sofre de problemas similares àqueles associados com a comunicação entre diferentes comunidades de informação [11]. O diferencial é que as entidades envolvidas não são pessoas capazes de executar abstração e senso de raciocínio comum sobre os significados dos termos, mas máquinas. No entanto, aplicações que usam diferentes ontologias para descrever os seus domínios ainda assim necessitam de interoperabilidade. Todavia, para que isso ocorra é necessário encontrar correspondências entre as ontologias, ou seja, dadas duas ontologias, sermos capazes de notar as

similaridades e expressar estas correspondências de modo que a máquina possa processá-los.

Neste artigo, é proposto um selecionador de recursos baseado em múltiplas ontologias, ontologias estas que representam as visões que cada OV da grade tem sobre os seus recursos. Com este selecionador espera-se atender solicitações de usuários ou aplicações de forma transparente. Em outras palavras, os usuários não precisam conhecer como foram descritos os recursos em uma grade.

O artigo é organizado da seguinte forma: na seção 2, são apresentados brevemente alguns aspectos relativos às ontologias. A seção 3, de forma detalhada apresenta os trabalhos relacionados. A proposta, os principais componentes da arquitetura do sistema selecionador proposto e a ontologia criada são descritos na seção 4. Na seção 5, são descritos o ambiente experimental e os estudos de caso realizados. Finalmente, na seção 6, conclui-se o trabalho juntamente com perspectivas de trabalhos futuros.

2 Ontologia

Ontologia de domínio pode ser entendida como uma descrição definindo vocabulários representativos deste domínio [12]. Como exemplo, definições são associadas a nomes de entidades dentro de um domínio compartilhado (classes, relações, funções e outros objetos) com um texto legível para um humano. Pretende-se através desses nomes fornecer significado ao domínio de universo estudado criando juntamente axiomas formais para restringirem a interpretação e o uso desses vocabulários bem formados.

2.1 Integração Semântica

Uma importante área de aplicação das ontologias é a integração de sistemas e bases de dados existentes [11]. A capacidade de troca de informação em tempo de execução, também conhecida como interoperabilidade no contexto de aplicação de ontologias, é um tópico importante. Contudo, a tentativa de fornecer interoperabilidade traz problemas similares àqueles associados com a comunicação entre diferentes comunidades de informação.

Portanto, para permitir que máquinas se entendam, é necessário explicar o contexto de cada sistema.

Uma direção para alcançar interoperabilidade entre dois diferentes sistemas é executar mapeamentos de ontologias para encontrar correspondências semânticas entre elas. No entanto, como diversos trabalhos de pesquisa indicam [11, 17, 8, 14], abordagens automáticas de mapeamento semântico entre ontologias não conseguem identificar a maioria das correspondências semânticas, ocorrendo conseqüentemente falta de interoperabilidade. Erigh & Sure [8] comentam ainda que um mapeamento totalmente automático pode levar a resultados incorretos.

Existem três diferentes abordagens de integração semântica de ontologias [24, 11], são elas:

- **Abordagem Centralizada:** cada fonte de informação está relacionada a uma única ontologia de domínio comum;
- **Abordagem Descentralizada:** toda fonte de informação está relacionada a sua própria ontologia;
- **Abordagem Híbrida:** toda fonte de informação tem sua própria ontologia, mas o vocabulário destas ontologias está relacionado a uma ontologia comum.

Em [3], é mencionado que a abordagem Híbrida tem a vantagem de não limitar a diversidade de modelos que descrevem um mesmo domínio, pois cada modelo possui a sua própria ontologia particular. Em adição, este paradigma não requer procedimento complexo no mapeamento de ontologias, pois somente os termos definidos em diferentes ontologias são relacionados semanticamente com termos de uma ontologia global compartilhada.

3. Trabalhos Correlatos

O uso de ontologias em ambientes de grade é relativamente recente. O emprego de ontologias se deve ao fato de poderem expressar e integrar fontes de informações heterogêneas, além de que atualmente há muitos dados e informações e pouco sabemos a respeito deles [6]. A seguir, são descritos os dois trabalhos que este se baseou junto com a contribuição desejada deste trabalho.

3.1. Ontology-based Matchmaker

Tangmunarunkit et. al propuseram em [23] um selecionador de recursos baseado em ontologias para selecionar recursos de grade com o objetivo principal de tornar o processo flexível e extensível conforme novas características e conceitos fossem acrescentados no ambiente.

Este objetivo é alcançado elaborando duas ontologias distintas para declarativamente descrever os recursos e os seus pedidos. Regras de inferência são utilizadas para permitir uma comparação semântica entre as ontologias.

Este fato faz com que o selecionador de recurso possa ser facilmente estendido, bastando acrescentar vocabulários nas ontologias e regras de inferência que permitiram realizar a comparação semântica. Além disso, implementa um sistema bastante interessante para detectar inconsistências nos pedidos enviados ao selecionador. Este sistema também se baseia em regras de inferência para não sobrecarregar o sistema de seleção de recursos com pedidos incoerentes.

Os autores deste trabalho consideram que as regras são elaboradas pelo(s) próprio(s) desenvolvedor(es) da ontologia de recursos (que descreve recursos da grade) e pedidos (que descreve os pedidos dos recursos), tornando transparente aos usuários do sistema *matching* como foram modelados ontologicamente os recursos da grade e como elaborar as regras de verificação de consistência.

Apesar de Tangmunarunkit et. al afirmarem que as OVs que compõem as grades são autônomas e independentes na descrição dos seus recursos, o selecionador proposto não trabalha com diversas visões (ontologias de recursos). Ao invés disso, consideram uma única ontologia de recursos para descrever todos os recursos na grade.

3.2. Resource matching baseado na união de ontologias

Em [15] foi desenvolvido um mecanismo de *matching* semântico que considera a união de várias ontologias de recursos em uma única ontologia denominada de ontologia global. A união dos conceitos equivalentes entre essas ontologias é alcançada através de um dicionário de sinônimos.

A inclusão destes sinônimos é feita utilizando a propriedade *owl:equivalentClass*, presente na linguagem de construção de ontologias OWL [18], que é empregada para indicar que os conceitos são equivalentes.

A busca de recursos sobre a ontologia global criada é executada pelo usuário através de consultas elaboradas na linguagem RDQL (RDF Data Query Language) [22].

Apesar desta abordagem considerar a autonomia e independência das OVs, contudo não traz facilidades aos futuros usuários do sistema de *matching* semântico. Para que as consultas sejam efetivas e realizadas sobre informações consistentes, os usuários necessitam estar familiarizados com a representação da sintaxe abstrata da linguagem OWL e traduzí-la para grafos RDF [16], pois a construção das consultas em RDQL é baseada nestes grafos.

Como citado em [14], não é de se esperar que qualquer usuário lide facilmente com o formalismo empregado para representar a base de conhecimento de um domínio qualquer. Somente usuários especialistas têm conhecimento para interagir com sistemas de conhecimento.

Outra característica desse mecanismo é a utilização de regras para manter a consistência das descrições dos recursos após a geração da ontologia global. Nesta abordagem, as regras de consistência devem ser introduzidas no sistema pelos usuários. Portanto, esta abordagem pressupõe que os usuários conheçam a sintaxe da linguagem de regras utilizadas pelo motor de inferência fornecido pela API Jena [2] bem como a sintaxe abstrata da linguagem OWL e a sua tradução para grafos RDF.

3.3 Contribuição

O presente trabalho de pesquisa considera múltiplas ontologias no processo de seleção de recursos na grade. Ou seja, entende-se que as OVs que compõem uma grade têm autonomia para descrever seus recursos. O diferencial desta contribuição é fornecer um processo de comparação semântica de recursos que considere esta autonomia e que não requeira dos usuários o conhecimento de como os recursos foram descritos. Esta facilidade é alcançada criando uma ontologia global de referência. Esta ontologia permite aos desenvolvedores das ontologias das OVs estabelecerem regras de mapeamento, tornando suas ontologias interoperáveis no sistema. Na próxima seção é apresentado o sistema de seleção proposto de forma mais detalhado.

4. Sistema Selecionador Proposto

Nesta seção, nós apresentamos o sistema selecionador de recursos de grade proposto. O sistema foi construído visando trabalhar com diversas ontologias de recursos da grade. Estas ontologias representam a visão que cada OV tem com relação aos seus recursos compartilhados na grade. O tipo de abordagem adotado para integrar as ontologias é a híbrida. A escolha deve-se ao fato de esta abordagem preservar as distintas visões das OVs sobre os seus recursos, ser escalável e facilitar a interoperação das ontologias, pois o processo de mapeamento é menos complexo quando comparado à abordagem descentralizada.

Em nossa proposta, os termos (isto é, nomes dados às classes e às suas propriedades) empregados na ontologia de recurso de cada OV são mapeados aos termos da Ontologia Global (OG), onde são descritos os termos ou conceitos referentes aos recursos da grade. Desenvolvemos a OG utilizando termos que caracterizam os recursos comumente encontrados em uma grade computacional. Estes termos não são tidos como padrões para caracterizar os recursos da grade, contudo servem de referência aos desenvolvedores das ontologias das OVs, como ilustrado no item(a) da Figura 1. Baseado na OG, cada desenvolvedor cria regras semânticas que mapeiam os termos da ontologia da OV com os termos da OG. Além disso, a OG tem o papel de uma linguagem de consulta, como apresentado no item (b) da Figura 1. Portanto, o usuário pode pesquisar recursos na grade atribuindo valores aos termos da OG (tornando-se requisitos) e submetê-los como consulta ao selecionador para que este escolha os recursos de acordo com os requisitos definidos.

O emprego da OG como linguagem de consulta se deve a dois fatores. O primeiro, é fato de os próprios termos da OG já estarem mapeados aos outros termos das demais ontologias que descrevem os recursos das OVs. Dessa

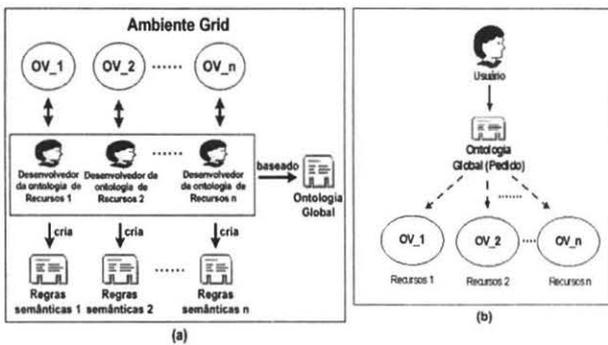


Figura 1. Abordagem de integração das ontologias de recursos de diversas OV's

forma, fica transparente ao usuário a seleção de recursos descritos em diversas ontologias. Em segundo lugar, o usuário não necessita ter profundos conhecimentos sobre o formalismo empregado para descrever uma consulta de recursos efetiva.

Para desempenhar o papel de linguagem de consulta, foram definidas adicionalmente na OG diretivas que indicam ao selecionador o tipo de seleção a ser realizada. Como exemplo, temos as diretivas *number_resources_return* e *rank_by*. O significado dessas diretivas serão explicadas posteriormente nos estudos de casos de seleção de recursos descritos na seção 5.

4.1. Arquitetura do Sistema Seleccionador

Os três principais componentes da arquitetura apresentam as seguintes características:

- **Ontologias de Domínio:** os domínios descritos foram dos recursos de grade e da OG. As ontologias do domínio de recursos foram projetadas para descrever os recursos de cada OV de acordo com suas visões. A OG foi desenvolvida para servir de referência às ontologias de recursos das OV's e como uma linguagem de consulta para o usuário do sistema.
- **Regras de verificação de inconsistência:** este componente captura as inconsistências nas consultas que contrariam os axiomas presentes na OG além de inconsistências no nível de instância, por exemplo, que uma consulta requerendo recurso com sistema operacional 'Windows' e ao mesmo tempo com sistema de arquivos 'EXT3' não é possível. As inconsistências são expressas por regras. Cada regra é formada por premissas, que representam uma situação inconsistente na consulta, e uma conclusão

que são mensagens informando a inconsistência, como apresentado na Figura 2.

```
<?xml version="1.0" encoding="UTF-8" ?>
<validator_request>
<set_rules>
<rule>[[?A req:file_system_type ?FS] (?B req:os_type ?OS) equal(?FS 'EXT3') equal(?OS 'Windows')] -> (?B conflict Operating_System_'Windows'_with_File_System_'EXT3'_)]
</rule>
<rule>[[?A req:min_physical_memory_available ?value] le(?value 0) -> (?A error value_attributed_to_'min_physical_memory_available'_requisite_must_be_greater_or_equal_zero.))
</rule>
.....
```

Figura 2. Exemplo de regras de inconsistência

- **Regras Semânticas:** neste componente é estabelecido correspondências semânticas entre características de um recurso expresso em uma determinada ontologia da OV com requisitos definidos na OG.

```
<?xml version="1.0" encoding="UTF-8" ?>
<regra_comparacao>
<tipa>
<requisito_pedido>[requisito_pedido]
<caracteristica_recurso>[os_type]
<regra_comparacao>[[?X req:os_type ?OS] (?Y grsc:os_type ?OSS) equal(?OS ?OSS) (?Z grsc:running_os ?Y) (?Z grsc:ip_address ?IP) -> (?Z grsc:result_ip ?IP)]
</regra_comparacao>
</tipa>
<requisito_pedido>[min_physical_memory_available]
<caracteristica_recurso>[free_physical_memory]
<regra_comparacao>[[?X req:min_physical_memory_available ?value1] (?Z grsc:free_physical_memory ?value2) get(?value2 ?value1) (?M grsc:running_os ?Z) (?M grsc:ip_address ?IP) -> (?M grsc:result_ip ?IP)]
</regra_comparacao>
.....
```

Figura 3. Regras definidas por uma OV

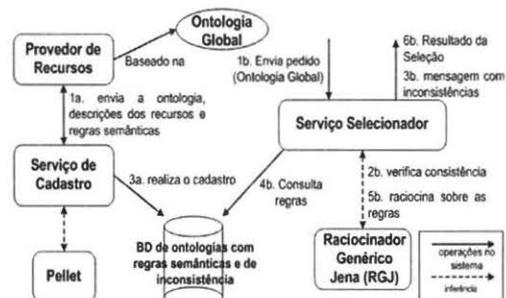


Figura 4. Arquitetura do sistema selecionador de recursos proposto

O sistema selecionador considera que tanto as OV's quanto o selecionador conhecem a OG. Este último também tem acesso às regras de verificação de inconsistência. Para publicar as suas ontologias, as OV's informam os mapeamentos semânticos entre os termos das suas ontologias de recursos com os da OG. Os mapeamentos e os seus significados são codificados nas regras semânticas,

como mostrado na Figura 3. A arquitetura do sistema selecionador, ilustrada na Figura 4, apresenta as seguintes seqüências de interação no sistema proposto:

- Cada OV (representado pelo Provedor de Recursos) envia a sua ontologia de recursos, as descrições dos recursos gerenciados e as regras semânticas para o Serviço de Cadastro (1a);
- O Serviço de Cadastro verifica a consistência das descrições dos recursos enviadas sobre a ontologia informada (2a), usando o raciocinador Pellet. Havendo inconsistências, ela é notificada à OV para poder corrigi-las e as descrições inconsistentes não são publicadas no banco de dados. As regras semânticas são também armazenadas no banco de dados (3a);
- Um usuário que deseja solicitar recursos ao selecionador deve utilizar a OG para criar consultas, definindo quais requisitos os recursos da grade devem satisfazer (1b);
- O selecionador ao receber uma consulta verifica a sua consistência baseado nas regras de inconsistência, previamente armazenadas, utilizando o raciocinador RGJ (2b). Caso encontre inconsistências, estas serão retornadas ao usuário como mensagens para permitir a correção da consulta (3b);
- Não havendo inconsistências o selecionador verifica quais requisitos foram informados na consulta. Logo a seguir, busca no banco de dados (4b) todas as regras semânticas das OVs que tem os mapeamentos com estes requisitos. Junto são trazidas as descrições dos recursos publicadas pelas OVs que são carregadas na memória principal. Estas descrições são representadas na memória por um único modelo Jena;
- Para cada requisito na consulta, o selecionador verifica quais regras relacionam as características dos recursos a este requisito e as envia ao raciocinador RGJ para que as avalie sobre o modelo Jena criado (5b). A avaliação resulta em um modelo que representa os recursos que satisfazem o requisito analisado. Este procedimento é realizado para os demais requisitos, gerando outros modelos. Ao final, o selecionador faz a intersecção destes modelos para encontrar quais recursos atendem ao mesmo tempo todos os requisitos definidos pelo usuário na consulta;
- De acordo com as diretivas expressas na consulta o selecionador sabe quantos e quais recursos (que satisfazem todos os requisitos) deverão ser retornados como resultado do processo de seleção (6b).

4.2. A Ontologia Global

A Ontologia Global, ilustrada na Figura 5, foi desenvolvida com a principal funcionalidade de servir como uma ontologia de referência comum. Esta referência auxilia na interoperação das diferentes ontologias.

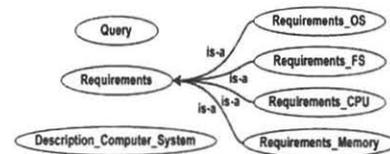


Figura 5. Hierarquia conceitual da Ontologia Global

Com o intuito do sistema selecionador proposto assimilar essas equivalências no processo de seleção, optou-se em utilizar como linguagem de consulta a própria OG. A OG foi descrita no idioma Inglês por ser praticamente um idioma universal, permitindo o entendimento por outros desenvolvedores de ontologias de recursos. Portanto, considerou-se que o processo de mapeamento dos conceitos não é afetado caso as ontologias das OVs tenham sido projetadas em diferentes idiomas.

5. Ambiente Experimental

Nos testes experimentais simulou-se uma grade computacional composta de três OVs denominadas de OV_1, OV_2 e OV_3. Cada OV tem sua própria ontologia para descrever os seus recursos. Considerou-se os recursos mais comuns para todos os experimentos. No entanto, a OG tem a capacidade de ter descrições de recursos mais específicos, bastando estendê-la.

A ontologia OV_1, ilustrada no item (a) da Figura 6, foi desenvolvida por nós, enquanto as outras duas, OV_2 (item (b)) e OV_3 (item (c)), foram baseadas nos respectivos trabalhos de pesquisa [19] e [21]. Vale ressaltar as distintas visões das OVs, através das diferentes estruturas hierárquicas que definem os seus recursos conceitualmente. Adicionalmente, os termos atribuídos aos conceitos presentes nas ontologias empregam idiomas distintos, nesta demonstração o inglês e o português.

Parte das informações dos recursos e suas políticas foram adquiridas através de consultas a uma configuração de grade real. Consultas foram executadas sobre um total de trinta recursos computacionais. Cada OV, antes de publicar dez dos seus recursos, informou as equivalências semânticas da sua ontologia ao sistema. As ontologias de recursos das OVs e a global foram construídos na linguagem OWL, através do editor Protégé.

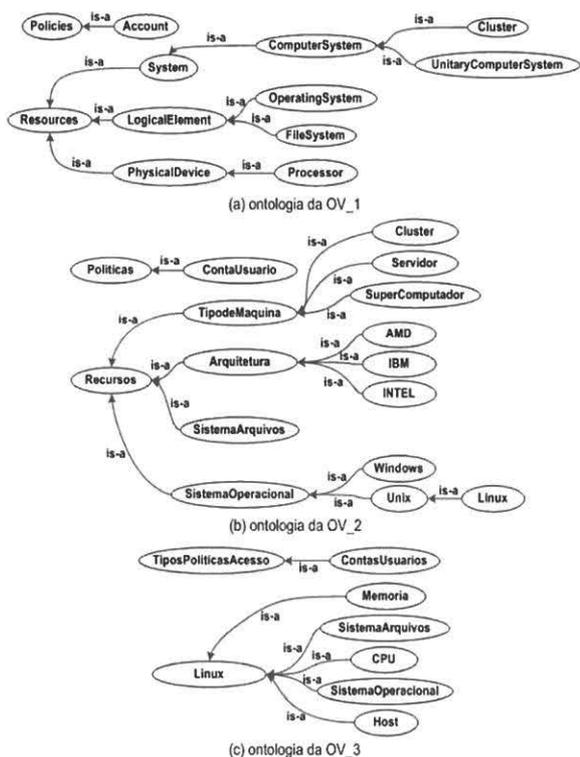


Figura 6. Visão das OV's sobre seus recursos

Os estudos de caso foram realizados sobre as máquinas descritas na Tabela 1. A máquina 1 (M1) agiu como cliente, ou seja, de onde as consultas foram elaboradas e submetidas. A máquina 2 (M2), foi onde o serviço selecionador foi posto a executar.

Tabela 1. Configuração das máquinas

Máquina	Sist. Operacional	Memória	Processador
M1	Windows XP	256 MB	Duron (1200 MHz)
M2	Debian 3.1	512 MB	Pentium IV (1800 MHz)

5.1 Estudos de Caso

Na Figura 7, apresenta-se a interface criada para que o usuário possa elaborar sua consulta facilmente. Os estudos de caso foram criados sobre esta interface. Usando-a, um usuário pode indicar os requisitos que os recursos da grade devem atender bem como as diretivas de seleção que o selecionador deve seguir. Na primeira coluna, os usuários escolhem quais características dos recursos serão restringidas e as diretivas desejadas. Através da terceira coluna, eles atribuem os valores às características

e diretivas escolhidas. Na segunda coluna o usuário pode buscar os metadados que explicam o significado dos termos (características de recursos e diretivas) definidos na ontologia OG. Estes metadados foram definidos na própria OG, e auxiliam os usuários na definição de consultas efetivas.

No primeiro estudo de caso realizado, visamos demonstrar a facilidade de definir consultas, além da realização de *matching* semântico. A consulta *pedido_6*, ilustrada na Figura 7 requer recursos com as seguintes características: tenham sistema operacional Linux, no mínimo dois processadores de 1800 MHz, quantidade de memória RAM e disco mínimo disponível de 20GB e 190MB, e que possa ser acessado pelo usuário 'parra'. Em adição, foram definidos diretivas que indicam ao selecionador que dentre os recursos que atendam todas as restrições é para retornar os dois recursos que têm a maior capacidade de processamento.



Figura 7. Interface de edição de consultas

A definição da consulta *pedido_6* em SPARQL [20], mostrada na Figura 8, tem como objetivo mostrar o grau de conhecimento requerido dos usuários para construir consultas efetivas. Para isso, o usuário necessita conhecer as estruturas das ontologias e como representá-las através de triplas para alcançar as descrições dos recursos, representadas na cláusula *WHERE*. As restrições sobre as descrições alcançadas são definidas na cláusula *FILTER*. Portanto, nota-se que não é qualquer usuário que tem condições de elaborar consultas SPARQL. Optou-se em mostrar a representação do *pedido_6* em SPARQL, pois esta linguagem é mais expressiva que a RDQL. Contudo, o grau de conhecimento para definir consultas em RDQL é idêntico a SPARQL. O *pedido_6* foi definido para consultar somente recursos descritos na OV_1.

No lado esquerdo da Figura 9 são mostrados os requisitos presentes na consulta *pedido_6*, com destaque aos valores atribuídos às diretivas de seleção, *number_resources_return* e *rank_by*. No lado direito, são apresentadas as características dos dois recursos retornados como resultado da seleção. Destacamos a capacidade dos

```

PREFIX grsc: <http://www.owl-ontologies.com/OV_1.owl#>
SELECT ?ip, ?so, ?cpus, ?capProc, ?ram, ?disco, ?usuario
WHERE {
  ?A grsc:ip_address ?ip .
  ?A grsc:number_of_cpus ?cpus .
  ?A grsc:running_os ?B .
  ?A grsc:has_processor ?C .
  ?A grsc:running_os ?D .
  ?A grsc:has_file_system ?E .
  ?A grsc:authorized_account ?usuario .
  ?B grsc:os_type ?so .
  ?C grsc:processor_capacity ?capProc .
  ?D grsc:available_physical_memory ?ram .
  ?E grsc:available_space ?disco .
FILTER ( (?so = 'Linux') && (?cpus >= 2) && (?capProc >= 1800) &&
  (?ram >= 190) && (?disco >= 20) && ?usuario = 'parra' ) }
ORDER BY ?capProc
LIMIT 2

```

Figura 8. Pedido.6 definido em SPARQL

processadores, o primeiro de 3.200 MHz e o segundo de 2.400 MHz. Os recursos identificados por *Host.8* e *Cluster.3* pertencem respectivamente às ontologias *OV.3* e *OV.2*. Dentre os trinta recursos disponíveis na grade, estes recursos satisfazem todas as restrições definidas na consulta *pedido.6*, além de possuírem a maior capacidade de processamento entre os recursos que atendem a consulta.



Figura 9. Consulta baseada em semântica

Nota-se que os termos usados para identificar a capacidade dos processadores nos dois recursos (*clockCPU* e *velocidade.cpu*) são distintos, contudo são considerados semanticamente equivalentes no sistema. Isso ocorre, pois tanto a *OV.3* e *OV.2* ao definirem as equivalências dos conceitos de suas ontologias com os da OG informaram respectivamente que o termo *clockCPU* e *velocidade.cpu* são equivalentes ao termo *min_clock_speed* presente na OG.

O segundo estudo de caso realizado deseja mostrar a característica de verificação de consistência que todas as consultas são submetidas no sistema selecionador proposto.

A segunda consulta enviada ao serviço selecionador apresentou três inconsistências, detectadas com o auxílio das regras e do RGJ, como mostrado na Figura 10. Ao enviar esta consulta foi retornado uma mensagem mostrando as inconsistências, quais são:

- o valor atribuído ao requisito *min_physical_memory_available* deve ser maior ou igual a zero;
- Sistemas operacionais da família *Windows* não suportam o sistema de arquivos *EXT3*;
- o valor atribuído à diretiva *rank_by* foi *min_clock_cpu*. Contudo, o valor deve ser um dos próprios requisitos (de tipo numérico) definidos na OG. Neste caso em específico devia ser o termo *min_clock_speed*.

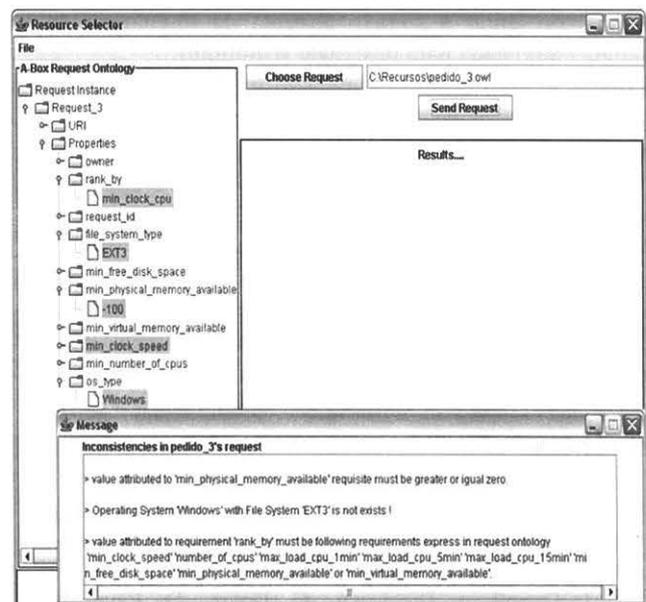


Figura 10. Pedido contendo inconsistências

6. Conclusão e Trabalhos Futuros

Neste artigo, foi apresentada uma proposta e um protótipo de um selecionador que objetiva reduzir os desafios relacionados à busca de recursos de grades de diferentes OVs. Esta proposta permite a seleção de recursos de grades computacionais baseado na interoperabilidade de múltiplas ontologias.

Para alcançar esse objetivo foi empregada a abordagem híbrida para integração de diferentes ontologias do domínio de recursos da grade. Esta abordagem foi

escolhida pois, dentre as abordagens de mapeamento de ontologias, possui menor complexidade, permitindo alta escalabilidade. Para descrever os mapeamentos, tornando as ontologias interoperáveis no sistema, foram empregadas regras semânticas que relacionam os termos expressos nas ontologias com os termos da OG. Também adotou-se regras para reconhecerem inconsistências nas consultas submetidas ao selecionador de forma a não sobrecarregar o sistema com consultas incoerentes. Ambos os tipos de regras são interpretadas por um motor de raciocínio. A utilização de regras e a possibilidade de inclusão e exclusão delas quando necessário no sistema tornam o processo de *matching* semântico proposto flexível.

O sistema desenvolvido usou a própria OG desenvolvida como linguagem de consulta para facilitar aos usuários a elaboração de consultas. Esta abordagem, diferentemente de pesquisas realizadas na área, torna transparente aos usuários como os recursos das grades foram descritos e como representar a sintaxe abstrata da linguagem da ontologia empregada para descrever o domínio.

Como trabalhos futuros, planeja-se ampliar as formas de consultas realizadas pelo selecionador desenvolvido. Com este objetivo, pretende-se estender a OG para que o usuário possa descrever expressões lógicas tanto conjuntivas quanto disjuntivas sobre os requisitos. Conseqüentemente, estenderemos o selecionador para que as reconheça e execute a combinação dessas expressões lógicas corretamente. Analisar o desempenho do selecionador estendido em termos do número de regras avaliadas pelo raciocinador e de recursos disponíveis na grade.

Referências

- [1] J. Brooke, D. Fellows, K. L. Garwood, and C. A. Goble. Semantic matching of grid resource descriptions. *European Across Grids Conference*, pages 240–249, 2004.
- [2] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: implementing the semantic web recommendations. In *13th World Wide Web Conference*, pages 74–83, 2004.
- [3] S. Casare and J. S. Sichman. Using a functional ontology of reputation to interoperate different agent reputation models. *JBCS*, 11(2):79–94, November 2005.
- [4] L. Chuang and I. Foster. A constraint language approach to matchmaking. *14th International Workshop on Research Issues on Data Engineering (RIDE)*, 00:7–14, 2004.
- [5] L. Chuang, L. Yang, I. Foster, and D. Angulo. Design and evaluation of a resource selection framework. *11th IEEE International Symposium on High-Performance Distributed Computing*, pages 63–72, 2002.
- [6] A. Congiusta, C. Mastroianni, A. Pugliese, D. Talia, and P. Trunfio. Enabling knowledge discovery services on grids. *European Across Grids Conference*, pages 250–259, 2004.
- [7] K. Czajkowski, I. Foster, C. Kesselman, V. Sander, and S. Tuecke. A protocol for negotiating service level agreements and coordinating resource management in distributed systems. *8th Workshop on Job Scheduling Strategies for Parallel Processing*, (2537):153–183, 2002.
- [8] M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *1st ESWS*, volume 3053, pages 76–91. Springer-Verlag, 2004.
- [9] I. Foster and C. Kesselman, editors. *The Grid: Blueprint for A New Computing Infrastructure*. Morgan Kaufmann Publishers, San Francisco, 1999.
- [10] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 3(15), 2001.
- [11] F. Freitas, H. Stuckenschmidt, and N. F. Noy. Ontology issues and applications. *JBCS*, 11(2):5–16, November 2005.
- [12] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, June 1993.
- [13] F. Heine, M. Hovestadt, and O. Kao. Towards ontology-driven p2p grid resource discovery. *5th IEEE/ACM International Workshop on Grid Computing*, pages 76–83, 2004.
- [14] Y. Kalfoglou and M. Schorlemme. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.
- [15] J. G. R. C. Lopes, A. C. M. A. Melo, M. A. R. Dantas, and C. G. Ralha. A proposal and evaluation of a mechanism for grid ontology merge. *High Performance Computing System and Applications*, 0:2, 2006.
- [16] B. McBride, K. Graham, and C. J. J. Resource description framework (rdf): Concepts and abstract syntax. Disponível: <http://www.w3.org/TR/rdf-concepts/#section-rdf-graph>, 2004.
- [17] N. F. Noy. Semantic integration: a survey of ontology-based approaches. *ACM SIGMOD Record, Special Issue on Semantic Integration*, 33(4):65 – 70, 2004.
- [18] P. F. Patel-Schneider and I. Horrocks. Owl web ontology language semantics and abstract syntax - mapping to rdf graphs. Disponível: <http://www.w3.org/TR/owl-semantic/mapping.html#4.1>, 2004.
- [19] A. M. Pernas and M. A. R. Dantas. Grid computing environment using ontology based service. *5th International Conference on Computational Science (ICCS'05), Lecture Notes in Computer Science*, 3516:858–861, May 2005.
- [20] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. *W3C*, June 2007. <http://www.w3.org/TR/rdf-sparql-query/>.
- [21] T. G. Ramos. Mecanismo extensível de descoberta de recursos em ambiente de computação em grade. 2005. Dissertação de Mestrado, Universidade de Brasília.
- [22] A. Seaborne. Rdfql - a query language for rdf. Disponível: <http://www.w3.org/Submission/RDFQL/>, 2004.
- [23] H. Tangmunarunkit, S. Decker, and C. Kesselman. Ontology-based resource matching in the grid - the grid meets the semantic web. *1st Workshop on Semantics in Peer-to-Peer and Grid Computing*, May 2003.
- [24] U. Visser, H. Stuckenschmidt, H. Wache, and T. Voegelé. Enabling technologies for interoperability. *Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, pages 35–46, 2000.