

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/260086669>

Tratamento Semântico de Conhecimento Organizacional em um Ambiente de Desenvolvimento de Software.

Conference Paper · January 2006

CITATIONS

2

READS

38

2 authors:



Fabiano Borges Ruy

Universidade Federal do Espírito Santo

22 PUBLICATIONS **193** CITATIONS

[SEE PROFILE](#)



Ricardo de Almeida Falbo

Universidade Federal do Espírito Santo

172 PUBLICATIONS **1,661** CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ODE: Ontology-based software Development Environment [View project](#)



Knowledge Management in Software Testing [View project](#)

Tratamento Semântico de Conhecimento Organizacional em um Ambiente de Desenvolvimento de Software

Fabiano Borges Ruy, Ricardo de Almeida Falbo

Departamento de Informática – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari s/n, Campus de Goiabeiras – 29.060-900 – Vitória – ES – Brasil

fabianoruy@yahoo.com.br, falbo@inf.ufes.br

Abstract. *More and more researchers recognize the importance of explicitly treating semantics in computer-based systems. In this context, Software Engineering Environments (SEEs) need to evolve to Semantic SEEs. This paper discusses the recent progress done in ODE (Ontology-based software Development Environment) to incorporate semantics to its information. An infrastructure adding inference capacities was developed, and we mention how it is being used to provide organizational knowledge management services in the environment.*

Resumo. *Cada vez mais a importância de se tratar semântica explicitamente em sistemas é reconhecida. Seguindo na mesma linha, Ambientes de Desenvolvimento de Software (ADSs) precisam evoluir para ADSs Semânticos. Este trabalho discute os avanços feitos no ambiente ODE para tratar semanticamente suas informações, apresentando uma infra-estrutura para incorporar capacidades de inferência ao ambiente e seu uso para prover serviços de gerência de conhecimento organizacional.*

1. Introdução

Os estudos na área de semântica aplicada a sistemas de software vêm crescendo nos últimos anos. Abordagens promissoras para tratar problemas no desenvolvimento de aplicações e para aperfeiçoá-las (dentre elas a Gerência de Conhecimento e a Web Semântica) compartilham uma percepção da importância da semântica [McComb, 2004]. Esse é o caso também dos Ambientes de Desenvolvimento de Software (ADSs). ADSs, ao longo de sua história, têm mostrado por meio de várias de suas características (grande quantidade de informações, complexidade das tarefas, premissa de integração, apoio ao trabalho em equipe, manipulação de conhecimento etc.) uma forte necessidade de tratar semanticamente as suas informações [Falbo *et al.*, 2005].

Grande parte do sucesso de um sistema está no “o quê” os símbolos manipulados significam no mundo real e no grau em que as pessoas e outros sistemas compreendem e concordam com os significados aplicados. Isso é especialmente importante em ADSs, dado que, ao longo do processo de software, muitas informações são produzidas e requeridas por diferentes pessoas, sendo essencial estabelecer conexões entre recursos de informação para se obter um conjunto de informações capaz de apoiar a realização de uma atividade [Falbo *et al.*, 2004].

Para lidar com essa questão, ontologias podem ser utilizadas para estabelecer um entendimento comum sobre diversos domínios, tal como o de engenharia de software. Interconectando os recursos de informação de um ADS com meta-dados baseados em ontologias, é possível adicionar semântica a eles, o que abre espaço para que o ADS proveja um nível qualitativamente novo de serviços. Um ADS Semântico pode ser visto, então, como um ADS em que parte da informação manipulada possui um significado formal (semântico), aumentando a habilidade das ferramentas para trabalhar em cooperação umas com as outras e com os desenvolvedores [Falbo *et al.*, 2005].

No contexto do Projeto ODE (*Ontology-based software Development Environment*) [Falbo *et al.*, 2004], que visa ao desenvolvimento de um ADS Semântico, ontologias têm sido usadas no sentido de apoiar o tratamento semântico das informações. Uma arquitetura de meta-dados baseada em ontologias foi definida para o ambiente e suas classes são derivadas a partir das ontologias que fundamentam o ambiente [Falbo *et al.*, 2005].

Este trabalho discute a ampliação do uso de semântica em ODE por meio da utilização de mecanismos de inferência que se apóiam na arquitetura de meta-dados do ambiente, com foco no conhecimento sobre organizações de software. A seção 2 discute sucintamente a importância da aplicação de semântica em ADSs e apresenta o ambiente ODE. A seção 3 apresenta a ontologia de organizações de software definida para o Projeto ODE. A seção 4 apresenta a infra-estrutura semântica definida e discute brevemente sua aplicação na derivação de serviços de gerência do conhecimento sobre organizações. Por fim, a seção 5 apresenta as conclusões deste trabalho.

2. Semântica em Ambientes de Desenvolvimento de Software

Semântica é geralmente definida como o estudo do significado das palavras. Em uma visão mais ampla, porém, percebe-se que semântica é mais do que isso. Em sistemas computacionais, ela está diretamente relacionada a “o quê” os símbolos manipulados pela máquina realmente significam no mundo real, o que envolve não somente o significado dos termos, mas também o grau em que as pessoas e outros sistemas que lidam com os mesmos entendem e concordam com o significado atribuído.

Os sistemas não existem sem semântica. Ela está presente, mesmo que implicitamente, em todo o ciclo de vida do software. Requisitos levantados com o usuário, modelos criados, trocas de dados entre aplicações, mensagens apresentadas aos usuários e estímulos produzidos pelos usuários para que o sistema execute funções são exemplos de que é necessário um entendimento compartilhado dos símbolos ou palavras manipulados pelos sistemas ou pelas pessoas. Entretanto, o grau no qual os sistemas preocupam-se com semântica varia bastante. Sistemas que dão maior atenção à semântica podem se tornar mais úteis, na medida em que algumas tarefas podem passar a ser mais bem apoiadas pelos sistemas, usando como base o seu “conhecimento”. Exemplos disso são buscas, verificação da coerência de algumas ações dos usuários e sugestões oportunas por parte dos sistemas, dentre outros [McComb, 2004].

Ambientes de Desenvolvimento de Software (ADSs) têm uma história de cerca de duas décadas, iniciando com o apoio a pequenos fragmentos do processo de software, passando pela noção de ADSs Centrados em Processos e avançando para a incorporação de conhecimento sobre domínios de aplicação, sobre engenharia de software e a

introdução de serviços de gerência de conhecimento. Ao longo dessa história, é possível perceber diversos motivos pelos quais um maior apelo semântico sempre foi necessário aos ADSs. Nas ferramentas CASE, buscava-se integrá-las, fazendo com que trocassem, ao menos, dados. E esperava-se que os dados de uma ferramenta fossem compreendidos pelas outras com as quais interagiam. Os ADSs têm como essência a integração e, por conseguinte, a necessidade de um entendimento compartilhado dos conceitos utilizados. ADSs Centrados em Processos, por sua vez, precisam que as ferramentas compreendam os processos de software definidos. Por fim, os ADSs que incorporam conhecimento de domínio, de engenharia de software ou organizacional têm como fundamento compreender, mesmo que parcialmente, o conhecimento manipulado, para que possam utilizá-lo efetivamente [Ruy, 2006].

Em qualquer momento da história dos ADSs, a noção de integração é considerada essencial. Integração de ferramentas implica no nível de concordância entre as ferramentas e envolve diversas dimensões, como apresentação, dados, controle, processo e conhecimento. A dimensão de apresentação se refere à interação com o usuário e está diretamente relacionada ao grau com que pessoas e sistemas concordam com o significado das interfaces. A semântica está presente no entendimento tanto das informações apresentadas pelo sistema aos usuários, quanto nos comandos para que os usuários solicitem a execução de funcionalidades. As dimensões de dados e de controle também são extremamente dependentes de semântica, uma vez que as ferramentas devem possuir o mesmo entendimento a respeito das estruturas lógicas e de dados (integração de dados) e dos serviços providos pelas ferramentas e pelo ambiente (integração de controle). A dimensão de processo estabelece uma ligação entre as ferramentas e o processo de software definido e depende de semântica no sentido de que todas as ferramentas do ambiente precisam compartilhar um entendimento comum sobre o que é um processo de software e qual o processo de software empregado. Por fim, a semântica é fundamental também para a dimensão de conhecimento, já que esta refere-se à gerência do conhecimento capturado nos projetos de software e à oferta de apoio baseado em conhecimento aos desenvolvedores [Falbo *et al.*, 2005].

McComb (2004) aponta que semântica tem a ver com as seguintes questões: Como dar nome às coisas? Como formar categorias? Como garantir algumas restrições? Como fazer com que esses aspectos afetem os sistemas a serem construídos? Sendo que essas são exatamente as questões que as ontologias procuram responder.

A importância das ontologias para expressar semântica é reconhecida em várias áreas, tais como *Web Semântica* e *Gerência de Conhecimento* [Davies *et al.*, 2003]. Essas áreas têm em comum o problema de crescimento rápido e contínuo do volume de informações, tornando difícil a sua recuperação, organização, acesso e manutenção. Informações importantes estão normalmente espalhadas em vários repositórios e os usuários despendem muito tempo acessando-os e lendo-os, para então descobrir como esses recursos estão relacionados e onde eles se encaixam na estrutura geral do domínio do problema. Neste contexto, ontologias são muito úteis, dado que podem combinar o entendimento humano dos símbolos com a capacidade de processamento das máquinas. Além disso, o modelo de domínio descrito por uma ontologia pode ser considerado como uma estrutura unificadora, dando à informação uma representação comum e semântica [Davies *et al.*, 2003]. O uso de meta-dados, baseados em ontologias e capazes de serem manipulados pela máquina, é apontado como uma forma promissora de lidar

com o problema do rápido crescimento do volume de informações. Como “dados a respeito de dados”, meta-dados são um eficaz meio de incorporar semântica, pois podem armazenar o significado dos próprios dados que descrevem [McComb, 2004].

O uso de ontologias e meta-dados nelas baseados tem sido explorado no ADS ODE (*Ontology-based software Development Environment*) [Falbo *et al.*, 2004]. A premissa é a seguinte: se as ferramentas de um ADS são construídas baseadas em ontologias, a integração entre elas é facilitada, pois os conceitos envolvidos são bem definidos. Além disso, usando-se de esquemas de anotação conceitual e serviços de inferência, novas funcionalidades, mais poderosas, podem ser oferecidas.

Diversas ontologias do domínio da Engenharia de Software foram desenvolvidas para embasar a construção de ODE, dentre elas ontologias de processo de software, de qualidade de software e de artefatos de software. Essas ontologias são usadas para derivar as classes que implementam o ambiente, que são organizadas em três níveis, como mostra a Figura 1. O nível ontológico é responsável pela descrição das ontologias no ambiente e suas classes correspondem à meta-ontologia assumida no ambiente. Assim, nesse nível há classes como *Conceito*, *Relacao*, *Propriedade*, sendo que as instâncias da classe *Conceito* servem de meta-dados para objetos do nível de conhecimento e do nível base. O nível de conhecimento abriga as classes que descrevem o conhecimento em relação a um domínio de aplicação. Suas classes são derivadas das ontologias, mais precisamente a partir de instâncias da classe *Conceito*. Os objetos dessas classes atuam no papel de conhecimento sobre os objetos do nível base. Por fim, o nível base define as classes que implementam o ambiente. As ontologias exercem um papel fundamental em ODE, na medida em que são responsáveis por organizar e estabelecer as conexões entre itens distribuídos pelos três níveis.

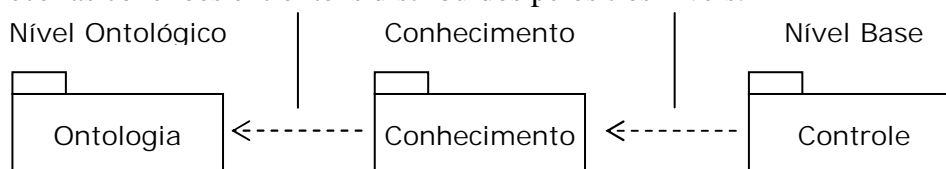


Figura 1 – Arquitetura Conceitual em 3 Níveis.

Anotações conceituais são utilizadas de duas formas em ODE: (i) por meio de uma associação *tipo* existente entre classes do nível base e do nível de conhecimento, e (ii) por meio dos atributos *classeConhecimento* e *classeBase* pertencentes à classe *Conceito*, do nível ontológico, que associam um conceito de uma ontologia à classe que o implementa, respectivamente, no nível de Conhecimento ou Base.

Uma característica que limita essa abordagem é sua forte dependência em relação à linguagem orientada a objetos utilizada para implementar o ambiente (Java), dado que axiomas de uma ontologia, por exemplo, podem ser manipulados mais adequadamente por meio de linguagens lógicas, principalmente quando mecanismos de inferências são utilizados. Visando buscar soluções mais apropriadas, definiu-se uma infra-estrutura para manipular de forma mais abrangente o conhecimento, oferecendo certas facilidades aos desenvolvedores do ambiente, de modo que serviços semânticos possam ser mais facilmente desenvolvidos e utilizados para apoiar tarefas no ambiente.

Antes de apresentar a infra-estrutura proposta (seção 4), apresentamos, a seguir, a ontologia de organizações de software usada para ilustrar o uso da infra-estrutura.

3. Uma Ontologia de Organizações de Software

A ontologia proposta tem como objetivos formalizar uma parcela do conhecimento sobre organizações de software, fornecer um vocabulário comum que possa ser utilizado para representar conhecimento útil sobre essas organizações e permitir que esse conhecimento seja tratado em ODE. Dentre os benefícios esperados com o desenvolvimento e utilização dessa ontologia estão: (i) estabelecimento de uma estrutura para tratar conhecimento organizacional em ODE; (ii) crescimento da base ontológica de ODE, por meio da integração com as outras ontologias que fundamentam o ambiente; e (iii) possibilidade de trabalhar semanticamente com conhecimento acerca da estrutura organizacional e do capital intelectual.

A ontologia de organizações foi desenvolvida com base nos trabalhos de [Lima, 2004], [Cota, 2003] e [Leão et al., 2004]. Além disso, a ontologia resultante está integrada a outras ontologias do ambiente, mais especificamente à ontologia de processo de software [Bertollo, 2006], da qual conceitos foram reutilizados. Foram considerados requisitos essenciais para essa ontologia: a estrutura da organização, sua interação com os projetos, papéis existentes, equipes de trabalho, distribuição das pessoas por unidades, cargos e projetos da organização e competências acumuladas por pessoas, requeridas por atividades ou demandadas para que se possa assumir um cargo. Assim, esse é o escopo da ontologia de organizações de software definida.

Em seu desenvolvimento foi adotado o método SABiO (*Systematic Approach for Building Ontologies*) [Falbo, 2004], cujas principais atividades são: (i) identificação do propósito e especificação de requisitos, que visa a identificar questões que a ontologia deve ser capaz de responder (questões de competência); (ii) captura da ontologia, que tem por objetivo capturar os conceitos, relações, propriedades e restrições relevantes sobre o domínio em questão, produzindo um modelo conceitual escrito em um perfil UML para modelagem de ontologias e definições textuais dos termos; (iii) formalização, que busca escrever a ontologia em uma linguagem formal, incluindo axiomas formais; (iv) integração com ontologias existentes, que visa a reutilizar conceituações existentes e integrar a ontologia em desenvolvimento a uma rede de ontologias mais ampla; (v) avaliação da ontologia, que, dentre outros, trata de avaliar se a ontologia é capaz de responder às questões de competência; e (vi) documentação da ontologia, que visa a registrar o desenvolvimento da ontologia.

Convém mencionar que na construção da ontologia de organizações de software, as atividades (ii) e (iv) se fundiram, pois duas características marcantes dessa ontologia são o fato dela ter sido desenvolvida a partir de outras ontologias e sua forte integração com a ontologia de processos de software. Dessa forma, o trabalho foi conduzido da seguinte forma: a partir dos propósitos e questões de competência definidos, procurou-se identificar elementos relevantes em outras ontologias, compilando-os em uma nova ontologia que atendessem aos requisitos propostos.

Dada a complexidade do domínio, a ontologia de organizações de software foi dividida em duas sub-ontologias: de Competências e de Estrutura Organizacional. A primeira compreende o conceito *Pessoa* e sua relação com o conceito *Competência*, indicando que pessoas acumulam competências. As competências, por sua vez, estão dispostas em uma hierarquia de sub-tipos, sendo especializadas em *Conhecimento*, *Experiência*, *Habilidade*, *Atitude* e *Valor*.

A Figura 2 mostra o modelo conceitual da sub-ontologia de Estrutura Organizacional. Como mostra esse modelo, uma organização é decomposta em unidades organizacionais, que, por sua vez, estão associadas entre si por relações de subordinação ou assessoria [Lima, 2004]. Em organizações de software, entidades que têm o capital intelectual como seu bem de mais alto valor, o conceito de *Pessoa* é central, estando relacionado a vários outros conceitos. Pessoas são contratadas para uma organização, onde assumem cargos, e são lotadas em unidades organizacionais. Além disso, pessoas acumulam competências e executam atividades nas equipes dos projetos a que estão alocadas. Cargos são ofertados em unidades organizacionais e demandam competências.

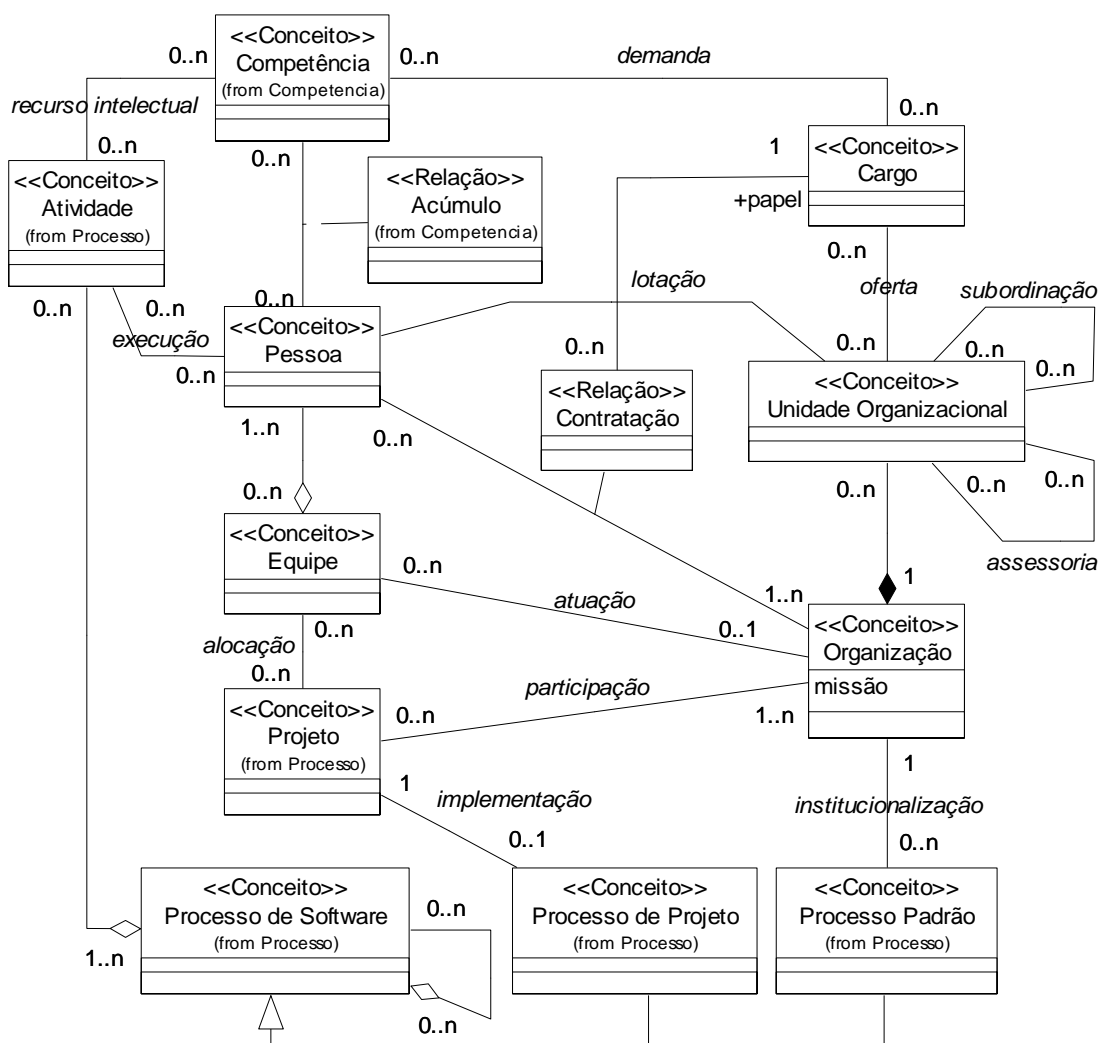


Figura 2 – Modelo Conceitual da Sub-ontologia de Estrutura Organizacional.

Devido à sua grande integração com a Ontologia de Processos de Software, há diversas relações entre as duas ontologias. O conceito de atividade está presente, uma vez que, para que atividades sejam executadas, são necessárias pessoas e requeridas competências. Pessoas compõem equipes que estão alocadas a projetos dos quais a organização participa. Por fim, uma organização institucionaliza processos padrão que podem ser instanciados em seus projetos.

Uma vez definida a ontologia de organizações, ela foi incorporada a ODE, criando-se as devidas instâncias no nível ontológico. Para a derivação das classes dos

outros dois níveis, foi utilizada, parcialmente, a abordagem de derivação de infra-estruturas de objetos a partir de ontologias, proposta em [Falbo *et al.*, 2002]. A Tabela 1 mostra parte do mapeamento feito. Os conceitos de competência e cargo foram mapeados para o nível de Conhecimento, pois tipicamente são usados para descrever características de objetos do nível base. Os demais conceitos foram mapeados diretamente em classes do nível Base, pois representam elementos concretos, reais, tais como organizações, pessoas, equipes etc., geralmente criados no contexto de algum projeto ou organização.

Tabela 1 – Conceitos da Ontologia e Classes de ODE Correspondentes.

Conceito / Relação	Classe Nível Conhecimento	Classe Nível Base
<i>Competência</i>	KCompetencia	-
<i>Pessoa</i>	-	RecursoHumano
<i>Acúmulo</i>	-	Acumulo
<i>Organização</i>	-	Organizacao
<i>Unidade Organizacional</i>	-	UnidadeOrganizacional
<i>Equipe</i>	-	Equipe
<i>Contratação</i>	-	-
<i>Cargo</i>	KRecursoHumano	-

Vale destacar que algumas restrições mais fortes foram colocadas no modelo de classes, sobretudo porque o ambiente foi concebido para ser usado por uma organização de cada vez. Assim, a relação *participação* é restringida, permitindo que os projetos do ambiente sejam conduzidos apenas por uma única organização. O mesmo vale para a relação *Contratação*, ou seja, pessoas trabalham apenas para uma organização, não sendo necessário, portanto, derivar uma classe para tratar essa relação.

4. A Infra-estrutura Semântica

Ainda que bastante úteis para o provimento de serviços no ambiente, o esquema de derivação de classes a partir de ontologias, com atributos e operações derivadas de propriedades e axiomas, ainda apresenta algumas limitações, sobretudo no que se refere ao uso de capacidades de inferência, conforme discutido na seção 2. Dessa forma, decidiu-se desenvolver uma infra-estrutura capaz de permitir inferência sobre o conhecimento armazenado. Para a construção dessa infra-estrutura, duas decisões tinham de ser tomadas: como representar o conhecimento e como manipulá-lo.

A linguagem OWL (*Ontology Web Language*) [Smith *et al.*, 2004] foi escolhida para representar as ontologias na infra-estrutura semântica de ODE. OWL permite representar explicitamente o significado dos conceitos, de suas relações, propriedades, restrições e indivíduos. Dada uma ontologia, a semântica formal de OWL especifica como derivar conseqüências lógicas, ou seja, fatos não literalmente presentes na ontologia, mas implícitos semanticamente [Smith *et al.*, 2004].

Para manipular o conhecimento representado pelas ontologias em OWL, foi utilizada a biblioteca Jena [McBride *et al.*, 2004], uma API Java usada na criação e manipulação de grafos RDF, incluindo OWL. O pacote *Jena 2 Ontology* possui classes para leitura e manipulação de ontologias. Jena oferece, também, o *OWL Reasoner*, uma máquina de inferência que provê suporte à realização de inferências sobre OWL DL.

Definidas a linguagem de representação e a biblioteca para realização de inferências, foi projetada a infra-estrutura, cujo processo é mostrado na Figura 3.

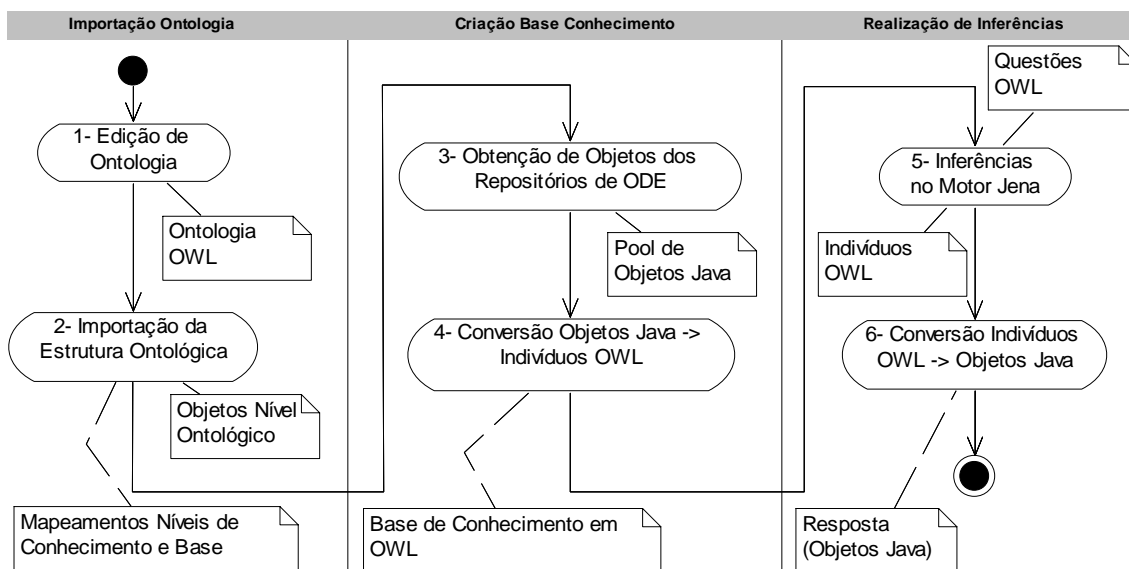


Figura 3 – Processo da Infra-estrutura Semântica.

Inicialmente, a ontologia deve ser incorporada ao ambiente, definindo-se os objetos do nível ontológico e suas ligações com os objetos dos outros dois níveis. Isso envolve os passos de edição da ontologia e importação de sua estrutura (1 e 2). Para a edição da ontologia foi utilizado o editor *Protégé-OWL* da ferramenta *Protégé*. O passo de importação da estrutura da ontologia é responsável pela incorporação da ontologia OWL nos níveis arquiteturais do ambiente. Os elementos da ontologia são importados para o ambiente, instanciando objetos no nível ontológico e estabelecendo suas ligações com os demais níveis. Por exemplo, uma classe OWL é importada como um objeto da classe *Conceito* do nível ontológico, que pode ser derivado em classes dos níveis de conhecimento e/ou base e essas ligações são mantidas através dos atributos *classeConhecimento* e *classeBase* de *Conceito*, conforme discutido na seção 2.

É importante apontar que o processo de criação das ligações entre os níveis não é feito de maneira completa. Conforme discutido na seção 2, os conceitos do nível ontológico são devidamente ligados às suas classes derivadas nos outros dois níveis. Entretanto, somente a ligação entre conceitos e classes não é suficiente para a criação de uma base de conhecimento completa. Desta forma, o passo 3 (Obtenção de Objetos a partir dos Repositórios de ODE) é capaz de obter, automaticamente, somente os objetos. Tomando por base os atributos *classeConhecimento* e *classeBase* da classe *Conceito*, é possível identificar as classes derivadas de cada conceito e obter seus objetos do repositório de dados do ambiente. Esses objetos são, então, incluídos em um *pool* de objetos para que sejam convertidos no passo 4. Mas as associações entre os objetos e os valores dos atributos não podem ser recuperados automaticamente, pois não há ligações entre as *relações* (nível ontológico) e as associações (demais níveis), ou entre as *propriedades* (nível ontológico) e os atributos (demais níveis). Para suprir a falta de ligações entre as relações e propriedades, durante o passo 4, é realizada uma varredura para que sejam obtidas as associações e atributos desejados, que são inseridos na base de conhecimento OWL resultante. A partir dessa base, é possível realizar os questionamentos desejados (passo 5).

Para encapsular o motor Jena, foi criada a classe `BaseConhecimento`, que possui os métodos `questionarModelo(relacao, valor)`, para submeter uma questão à máquina de inferência (passo 5) e `converterIndividuos(listaIndividuos)` para transformar as respostas (indivíduos OWL) em objetos Java (passo 6).

A infra-estrutura semântica desenvolvida foi utilizada para prover alguns serviços de gerência de conhecimento relacionados ao conhecimento organizacional, tomando por base a ontologia apresentada na seção 3, sendo o principal deles o serviço “quem sabe o quê” (mostrado na Figura 4), que passou a ser utilizado, por exemplo, na ferramenta de apoio à alocação de recursos humanos de ODE. Em sua forma original, a ferramenta sugeria quais pessoas poderiam executar uma atividade, baseando-se em seu tipo (classe `KRecursoHumano` do nível de Conhecimento, conforme mostrado na Tabela 1). Ou seja, se uma atividade precisava de um analista, a ferramenta sugeria todos os analistas da equipe. Utilizando o novo serviço, definido a partir da infra-estrutura, é possível realizar uma sugestão mais elaborada, baseando-se não apenas nos cargos ocupados pelas pessoas, mas também nas competências acumuladas por elas e requeridas pelas atividades. Esse serviço é capaz de listar um conjunto de pessoas mais adequado à atividade, baseando-se no axioma sobre a relação *execução*, que diz que, se uma pessoa acumula as competências requisitadas por uma atividade, então ela pode executar essa atividade.

```
/* Identifica quais pessoas acumulam a competência passada. */  
public List quemSabeOQue(KCompetencia competencia) {  
    return baseConhecimento.questionarModelo("acumulo", competencia);  
}
```

Figura 4 – Serviço “Quem Sabe o Quê”.

Outros serviços criados na infra-estrutura semântica, inferindo sobre a base de conhecimento da ontologia de organizações de software, incluem: (i) serviço de envio de perguntas, que, a partir de uma mensagem classificada segundo as competências relacionadas, pode enviá-la apenas aos destinatários que estariam aptos a respondê-la; (ii) serviço de aprovação de itens de conhecimento, que segue o mesmo princípio do anterior, distribuindo os itens de conhecimento registrados no ambiente (lições aprendidas, pacotes de mensagens) para que sejam aprovados por especialistas, reduzindo o acúmulo de itens nas mãos de um gerente de conhecimento; e (iii) serviço de planejamento de treinamento, que verifica quais pessoas não possuem um nível adequado de conhecimento em tópicos requeridos e as seleciona para treinamento.

5. Conclusões

A aplicação de semântica a sistemas tem sido alvo de diversos estudos e há vários trabalhos apontando benefícios da abordagem semântica em áreas como *Web Semântica* e *Gerência de Conhecimento* [Davies *et al.*, 2003]. Entretanto, Ambientes de Desenvolvimento de Software (ADSs), mesmo tendo muitas características comuns a essas outras áreas, ainda não têm explorado bem essa questão. Tratar explicitamente semântica em ADSs é um caminho a ser seguido em direção a ADSs Semânticos. Este artigo apresentou um passo nesta direção, por meio da definição de uma infra-estrutura que incorpora capacidades de inferência ao ambiente ODE, usando um esquema de anotação conceitual com meta-dados baseados em ontologias.

Agradecimentos

Este trabalho foi realizado com o apoio do CNPq e da CAPES, entidades do Governo Brasileiro dedicadas ao desenvolvimento científico e tecnológico, da FAPES, Fundação de Apoio à Ciência e Tecnologia do Espírito Santo, e das empresas VixTeam Consultoria & Sistemas e Projeta Sistemas de Informação, parceiras que têm financiado o projeto e dado feedback sobre sua aplicação a casos reais.

Referências

- Bertollo, G. (2006) Definição de Processos em um Ambiente de Desenvolvimento de Software. Dissertação de Mestrado, Mestrado em Informática, UFES.
- Cota, R.I. (2003) Um Estudo sobre o Uso de Ontologias e Padrões de Análise na Modelagem de Sistemas de Gestão Empresarial. Dissertação de Mestrado, Mestrado em Informática, UFES.
- Davies, J., Fensel, D., Van Harmelen, F. (2003) *Towards the Semantic Web: Ontology-driven Knowledge Management*, John Wiley & Sons.
- Falbo, R. A. (2004) “Experiences in Using a Method for Building Domain Ontologies” Proc. of the 16th International Conference on Software Engineering and Knowledge Engineering, International Workshop on Ontology In Action, Banff, Canada.
- Falbo, R.A., Ruy, F.B., Pezzin, J., Dal Moro, R., (2004) “Ontologias e Ambientes de Desenvolvimento de Software Semânticos”. 4th Ibero-American Symposium on Software Engineering and Knowledge Engineering, Vol I, Madrid, p. 277-292.
- Falbo, R.A., Ruy, F.B., Dal Moro, R. (2005) “Using Ontologies to Add Semantics to Software Engineering Environments”. 17th International Conference on Software Engineering and Knowledge Engineering, SEKE'2005, Taipei, China, p. 151-156.
- McComb, D. (2004) *Semantic in Business Systems: The Savvy Manager's Guide*. Morgan Kayfmann Publishers.
- Falbo, R.A., Guizzardi, G., Duarte, K.C. (2002) “An Ontological Approach to Domain Engineering”. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, SEKE'2002, Ischia, Italy, p. 351- 358.
- Leão, P.R.C., Oliveira, K.M., Moresi, E.A.D. (2004) “Ontologia de Competências Profissionais em Tecnologia da Informação”, Simpósio Brasileiro de Qualidade de Software, Workshop de Gerência de Conhecimento, Brasília.
- Lima, K.V.C. (2004) Definição e Construção de Ambientes de Desenvolvimento de Software Orientados a Organização. Tese de Doutorado, COPPE/UFRJ.
- McBride, B., Seaborn, A. (2004) Jena Tutorial, Jena 2 Ontology API, <http://jena.sourceforge.net/ontology/index.html>
- Ruy, F.B. (2006) Semântica em um Ambiente de Desenvolvimento de Software. Dissertação de Mestrado, Mestrado em Informática, UFES.
- Smith, M.K., Welty, C., McGuinness, D.L. (2004) OWL Ontology Web Language Guide. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>