Usando Ontologias, Serviços Web Semânticos e Agentes Móveis no Desenvolvimento Baseado em Componentes

Luiz H. Z. Santana, Antonio Francisco do Prado, Wanderley Lopes de Souza, Mauro Biajiz

Departamento de Computação (DC) – Universidade Federal de São Carlos (UFSCar)
Caixa Postal 676 – 13565-905 – São Carlos – SP

{luiz_santana, prado, desouza, mauro}@dc.ufscar.br

Abstract. *This paper presents an approach that combines Ontologies, Semantic Web Services and Mobile Agents, for the Component-Based Software Development. The Ontologies are employed to improve the problem domain analysis, and to get software components with a semantic description, which may be reused in a wide variety of applications. The Semantic Web Services are used as software components distributed over the Internet, and are composed to perform complex application tasks. The Mobile Agents manager the use of the Semantic Web Services, and can move through the network nodes in order to find, to composite and to monitor these services.*

Keywords: Component-Based Development, Ontologies, Semantic Web Services, Mobile Agents

Resumo. *Este artigo apresenta uma abordagem, que combina Ontologias, Serviços Web Semânticos e Agentes Móveis, para o Desenvolvimento Baseado em Componentes. As Ontologias são empregadas para aprimorar a análise do domínio do problema e para obter componentes de software com descrições semânticas, os quais podem ser reutilizados numa grande variedade de aplicações. Os Serviços Web Semânticos são utilizados como componentes de software distribuídos pela Internet e são compostos para realizar tarefas complexas de aplicações. Os Agentes Móveis gerenciam o uso de Serviços Web Semânticos e podem migrar através dos nós da rede a fim de encontrar, compor e monitorar esses serviços.*

Palavras-chave: Desenvolvimento Baseado em Componentes, Ontologias, Serviços Web Semânticos, Agentes Móveis

1. Introdução

Dentre as inúmeras pesquisas, que visam melhorar o Processo de Desenvolvimento de Software (PDS), destaca-se cada vez mais o uso de componentes de software. O Desenvolvimento Baseado em Componentes (DBC) é caracterizado pela integração, composição e adaptação de componentes pré-existentes, com ênfase no reuso de software, e apresenta vantagens tais como o aumento de produtividade e da qualidade de software [Lucrédio et al. 2004].

Um tópico importante de pesquisa no DBC é a construção de componentes suficientemente genéricos para serem reutilizados em diferentes aplicações. Uma

possível solução para esse problema é desenvolver um processo de construção de componentes para um domínio específico, baseando-se na análise desse domínio [Arango 1989]. Neste sentido, a Engenharia de Software introduziu recentemente o uso de ontologias, a qual possibilita a formalização do conhecimento, um melhor entendimento sobre o domínio e a sua representação num nível alto de abstração [Evermann e Wand, 2005].

Evolução dinâmica e não antecipada de software é um outro tópico importante no DBC, tendo em vista que os requisitos iniciais para a construção de aplicações podem ser modificados, causando assim alterações em seus componentes [Ebraert et al. 2005]. Uma possível solução para esse problema é o uso de Serviços Web Semânticos. Esses serviços encontram-se distribuídos pela Internet, podem ser modificados e usados pelas aplicações sem que seja necessário reconstruí-las. Entretanto, o uso desses serviços ainda oferece desafios: encontrar o serviço necessário; compor um conjunto desses serviços para atender a tarefas complexas; e monitorar um desses serviços ou uma composição dos mesmos, a fim de garantir o sucesso na execução da tarefa.

Freqüentemente Agentes de Software são empregados para facilitar o uso de Serviços Web Semânticos [Charif-Djebbar e Sabouret 2006, Lee 2007]. Interpretando as especificações desses serviços, Agentes de Software podem derivar conjuntos de regras simples, que definem o uso de Serviços Web Semânticos. Os agentes podem também migrar através dos nós de uma rede, a fim de lidar com a heterogeneidade das plataformas de hardware e software e com a distribuição dos serviços.

Em função do exposto, este trabalho apresenta uma abordagem para DBC, que combina Ontologias, Serviços Web Semânticos e Agentes Móveis. O uso de Ontologias visa à obtenção de componentes mais genéricos e padronizados, formalizando e incrementando a descrição do conhecimento do domínio de um problema. Os Serviços Web Semânticos possuem descrições semânticas que orientam o seu emprego nas soluções de problemas complexos que ocorrem na Internet. Os Agentes Móveis realizam a gerência, encontrando, compondo e monitorando a execução, do uso desses serviços, sendo que esses agentes podem migrar, através dos nós da rede, para utilizar serviços distribuídos pela Internet.

A seqüência desse artigo está organizada da seguinte forma: a seção 2 fornece uma visão geral de Análise de Domínio e Ontologias; a seção 3 versa sobre Serviços Web Semânticos, Agentes de Software e apresenta um framework que usa todas essas tecnologias; a seção 4 discorre sobre a abordagem proposta para DBC; a seção 5 descreve um estudo de caso, que foi desenvolvido empregando-se essa abordagem; a seção 6 compara trabalhos correlatos ao deste artigo; finalmente a seção 7 tece algumas conclusões e aponta para trabalhos futuros.

2. Análise de Domínio e Ontologias

O objetivo da Análise de Domínio é evitar que o processo de elicitação e codificação do conhecimento seja refeito para toda aplicação construída num mesmo domínio. Isso impede que o processo seja exposto a erros e inconsistências que já poderiam ter sido resolvidas, além de evitar perda de tempo, esforço e conseqüentemente recursos [Arango 1989]. Modelos de domínio, produtos da análise, são usados pela comunidade de reuso de software como especificações em alto nível de abstração. Estes modelos são

uma formulação genérica o suficiente para representar um conjunto de problemas, conhecimentos ou atividades similares. Uma das formas de obter e representar esses conhecimentos baseia-se em Ontologias.

Segundo [Gruber 1993], "Uma ontologia é uma especificação formal e explícita de conceitos compartilhados". A utilização de ontologias nessa etapa do processo de desenvolvimento traz vantagens como: melhoria da comunicação entre as pessoas envolvidas, uma vez que facilita a obtenção de um consenso sobre o vocabulário e os significados dos termos num domínio; formalização do conhecimento, já que a especificação do domínio em ontologias elimina contradições e inconsistências, resultando em especificações não ambíguas; e, principalmente, a representação do conhecimento para reuso, já que a ontologia descreve o conhecimento do domínio de forma explícita no seu mais alto nível de abstração. Possibilitando especializar o conhecimento durante o desenvolvimento de diferentes aplicações num domínio, que tenham propósitos variados, sejam criados por equipes distintas e em diferentes momentos.

Além disso, os modelos de ontologias apresentam vantagens em relação aos modelos tradicionais gerados pela análise de domínio (e.g., modelos de entidades e relacionamentos e modelos de objetos), já que estes são limitados para a representação de conhecimento, estabelecendo apenas significados particulares e estruturação para os conceitos envolvidos.

3. Serviços Web Semânticos e Agentes Móveis

Para aumentar a reutilização de software, pesquisas apontam, como passo fundamental, a sistematização do processo de análise e criação de componentes para um determinado domínio de aplicações [Werner e Braga 2000]. O Desenvolvimento Baseado em Componentes se preocupa com a criação de componentes que possam ser reutilizados em uma situação diferente daquela para qual foram originalmente construídos.

Componentes são unidades de software independentes que encapsulam seu projeto e implementação e oferecem serviços por meio de interfaces bem definidas para o meio externo. Por outro lado, Serviços Web são elementos computacionais autodescritivos e independentes de plataforma, que disponibilizam funcionalidades autocontida, visando o seu reuso e a interoperabilidade com outros sistemas [Papazoglou 2003].

Nesse contexto, um Serviço Web pode ser visto como um componente cujas funcionalidades são acessíveis por mensagens baseadas em *eXtensible Markup Language* (XML) [Yao e Etzkorn 2004, Ha e Lee 2006]. O arquivo de descrição do serviço, geralmente também baseado em XML, possui as informações necessárias para que outros componentes possam interagir com este serviço, incluindo o formato das mensagens para as chamadas os seus métodos, protocolos de comunicação e as formas de localização do serviço. Um dos maiores benefícios dessa descrição é a abstração dos detalhes de implementação do serviço, permitindo que seja acessado independente da plataforma de hardware ou software. A comunicação baseada em XML adiciona também flexibilidade com relação à linguagem de programação tanto na implementação, quanto no acesso ao Serviço Web. Estas características permitem e motivam a implementação de aplicações Web baseadas em Serviço Web por torná-las

fracamente acopladas com as outras partes do código da aplicação. Com isso, as aplicações adquirem uma arquitetura componentizada e flexível em relação às várias plataformas disponíveis no mercado.

Pela utilização de tecnologias da Web Semântica [Berners-Lee et al. 2001] é possível estender a capacidade dos Serviços Web tradicionais, criando os Serviços Web Semânticos. Esses serviços possuem descrições semânticas a fim de obter-se um maior poder de expressão na sua definição, na sua descoberta e no seu acesso [Hepp 2006]. Para realização de tarefas mais complexas uma composição de Serviços Web Semânticos pode usar agentes de software para escolher e combinar os serviços necessários. A execução de uma composição de serviços deve ser monitorada, a fim de detectar exceções e modificações dinâmicas que impeçam a concretização de uma tarefa a ser realizada. Isso permite que uma aplicação que dependa desses serviços se recupere, encontrando outros serviços ou criando uma nova composição que seja equivalente a anterior [Balzer et al. 2004].

Baseado em especificações providas por linguagens como a *Ontology Web Language for Services* (OWL-S) [W3Cb 2004], pode-se criar Agentes de Software, com comportamentos inteligentes e mobilidade para gerenciar o uso de Serviços Web Semânticos. Por sua vez, Agentes Móveis são Agentes de Software em execução num ambiente computacional, capazes de migrar de forma autônoma através dos elementos constituintes desse ambiente e compartilhar os seus recursos, a fim de realizar uma determinada tarefa [Dobson et al. 2006]. Tais agentes são particularmente usados em sistemas distribuídos, onde o poder computacional é descentralizado, pois contribuem para a interoperabilidade através dos elementos desses sistemas. Nesse trabalho, os Agentes Móveis foram escolhidos, uma vez que além de incorporar as funcionalidades dos agentes tradicionais, são capazes de migrar descobrindo, compondo e monitorando a execução dos serviços distribuídos pela Internet.

3.1. Framework para Serviços Web Semânticos, baseado em Agentes Móveis.

Buscando melhorar o processo de DBC e considerando as idéias apresentadas, foi desenvolvido um framework para apoiar a abordagem proposta. Esse framework, denominado FrameAgentesMóveis, combina Serviços Web Semânticos e Agentes Móveis, no desenvolvimento de aplicações a fim de facilitar sua implementação. No modelo da Figura 1 têm-se os principais componentes desse framework. O componente *Agent* é responsável pela: busca, composição e monitoramento da execução de Serviços Web Semânticos. Além disso, caso seja necessário, esse componente pode migrar atravésos nós da rede para encontrar os Serviços Web Semânticos necessários para realizar uma tarefa. As políticas que gerenciam a mobilidade são implementadas no componente *MobilityManager*, que deve ser reutilizado através de sua interface *IMobilityManager* , um dos pontos flexíveis do framework. Seu método *beforeMove* possibilita adicionar, além da necessidade de um Serviço Web Semântico remoto, situações em que o agente deve ou não migrar para realizar uma tarefa. Por exemplo, verificar se um recurso está disponível localmente ou remotamente. Já o método *beforeReceive* possibilita definir o comportamento no caso de receber agentes que migraram de outros nós da rede, podendo implementar os aspectos de segurança, desempenho e outros requisitos não funcionais.

O componente *Reasoner* é utilizado pelos agentes para efetuar inferências sobre as descrições do domínio e dos Serviços Web Semânticos. Os componentes *OntologyManager* e *ServicesRepository* são responsáveis pelo armazenamento e recuperação das descrições das ontologias em *Web Ontology Language* (OWL) [W3Ca 2004] e dos Serviços Web Semânticos em OWL-S, respectivamente.

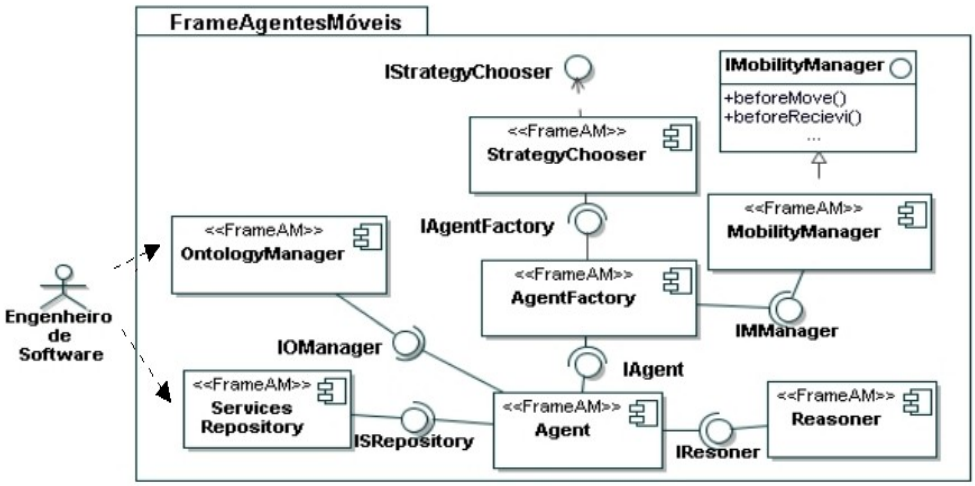

Figura 1. Modelo de Componentes do FrameAgentesMóveis.

O componente *StrategyChooser* foi construído com base no padrão *strategy* [Gamma et al. 1995] para suportar a escolha de diferentes estratégias na utilização dos Serviços Web Semânticos, conforme as necessidades de cada aplicação. Sua interface *IStrategyI* constitui outro ponto flexível do framework, e possibilita, por exemplo, que aplicação defina suas estratégias para adaptação de conteúdo, para comunicação em redes de sensores e integração em *WebLabs* [Coelho et al. 2007]. O componente *AgentFactory* foi construído segundo o padrão *factory* [Gamma et al. 1995], para suportar a criação de agentes conforme a estratégia escolhida no componente *StrategyChooser*.

A construção do FrameAgentesMóveis apóia a utilização da abordagem proposta neste artigo, servindo como um "esqueleto" que estrutura e organiza o desenvolvimento das aplicações, conforme se apresenta a seguir.

4. Abordagem Proposta

A abordagem proposta (Figura 2) é realizada nas quatro etapas do ciclo clássico de desenvolvimento de software (Análise, Projeto, Implementação e Testes). O processo de desenvolvimento é orientado por técnicas que atendem necessidades como a modificação não antecipada de componentes e a utilização de componentes distribuídos.

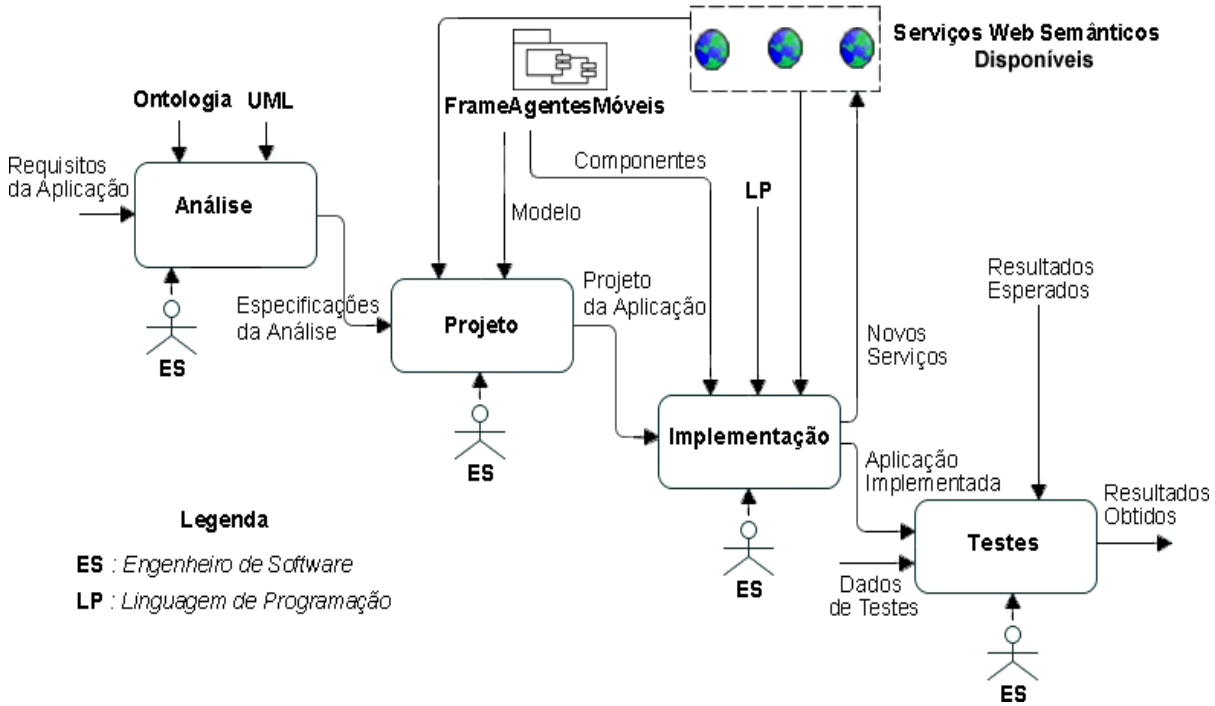

Figura 2. Abordagem proposta, segundo a SADT [Ross 1977].

Na etapa de **Análise** parte-se dos requisitos da aplicação para obter suas especificações em modelos de ontologias e outros modelos conhecidos da *Unified Modeling Language* (UML) [OMG 2004], como o de casos de uso e de seqüência. Os modelos de ontologias representam o conhecimento de maneira explicita, uma vez que estruturam os conceitos da aplicação num nível mais alto de abstração.

A partir das especificações da aplicação e da estrutura do FrameAgentesMóveis, na etapa de **Projeto**, faz-se a modelagem da aplicação considerando as restrições da plataforma de software adotada. Uma das atividades dessa etapa consiste em refinar os modelos de ontologias em modelos de classes. Além disso, são considerados os modelos especificados em UML e os modelos do framework para estruturar a utilização de Agentes Móveis e Serviços Web Semânticos. Nessa etapa também são projetados, os serviços que não estão disponíveis para reuso na Internet e os comportamentos relacionados com a mobilidade dos agentes.

Uma vez projetada a aplicação, faz-se a sua **Implementação**. São implementados os componentes específicos da aplicação e os pontos flexíveis do FrameAgentesMóveis (interfaces *IstrategyChooser* e *ImobilityManager*). São conectados também os Serviços Web Semânticos, conforme definido no projeto. Caso já estejam disponíveis na Internet, esses serviços podem ser reutilizados. Caso não existam, os novos serviços projetados, são implementados e disponibilizados para reuso. Em qualquer caso, o Engenheiro de Software adiciona as descrições semânticas desses serviços, em uma linguagem de marcação da Web Semântica, como a OWL-S utilizada nesse trabalho. A OWL-S possibilita a migração das descrições sintáticas em *Web Services Description Language* (WSDL) para descrições semânticas em OWL. Conforme ilustra a Figura 3, a OWL-S é organizada em três sub-ontologias:

ServiceProfile, que descreve as características, as capacidades do serviço e as transformações que esse serviço é capaz de realizar; **ServiceProcess**, que especifica o protocolo de interação com o serviço; e **ServiceGrounding**, que provê meios para que a comunicação com esse serviço seja feita através de mensagens *Simple Object Access Protocol* (SOAP).

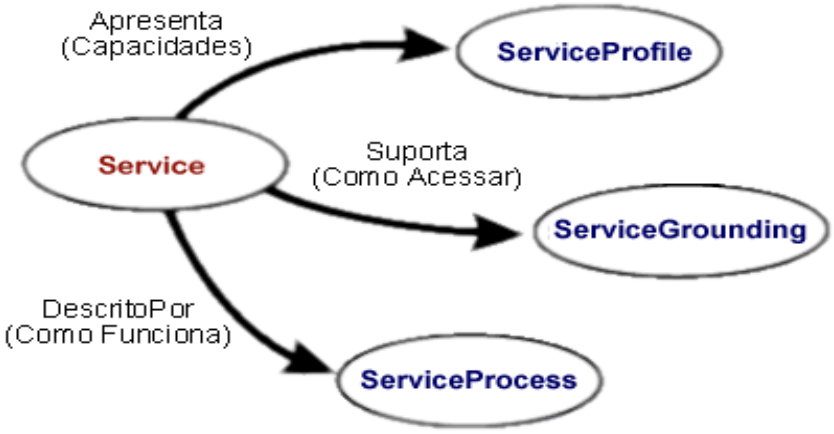

Figura 3. Visão Geral da OWL-S.

Finalmente, realizam-se T**estes** com dados que produzirão os resultados da execução. O Engenheiro de Software compara os resultados obtidos com os resultados esperados para verificar se atendem aos requisitos especificados. Caso não atenda retorna-se às etapas anteriores conforme o problema identificado, para as devidas correções.

5. Estudo de Caso

Para ilustrar o uso da abordagem proposta apresenta-se como estudo de caso uma aplicação no domínio da adaptação de conteúdo para computação ubíqua. A adaptação é realizada por um serviço de tradução de linguagens de marcação. Baseia-se em informações sobre o ambiente de uso de dispositivos móveis (e.g., preferências do usuário, rede de acesso, contrato com o provedor de serviços, características do dispositivo, conteúdo a ser adaptado). Essas informações são descritas em ontologias e denominadas perfis, a fim de que páginas da Web possam ser acessadas por esses dispositivos [Santana et al. 2007].

Para facilitar o entendimento dos requisitos dessa aplicação, as Figuras 4 e 5 fornecem uma visão geral da sua arquitetura. Na Figura 4 um Usuário acessa o serviço de adaptação de conteúdo através de um dispositivo móvel. Um Proxy de Adaptação tem o papel de interceptar requisições desse dispositivo para um Servidor de Conteúdo, que armazena conteúdo em seu formato original. Nesses Proxies estão localizados os Agentes Móveis, que utilizam Servidores de Adaptação disponíveis em Serviços Web Semânticos. Esses servidores realizam tradução de páginas *HyperText Markup Language* (HTML) para *Wireless Markup Language* (WML) e adaptações das imagens presentes nessas páginas.

Numa adaptação local, o fluxo de execução tem inicio após uma requisição de um usuário (1) ser interceptada por um Proxy de Adaptação. Em seguida, este Proxy de Adaptação requisita (2) e recebe (3), do Servidor de Conteúdo, o conteúdo a ser adaptado. De posse desse conteúdo, o Proxy de Adaptação delega ao seu agente a tarefa de realizar a adaptação necessária. Baseado-se na requisição e nos perfis, esse agente encontra, compõe e monitora a execução (4) dos Serviços Web Semânticos necessários. Finalmente, o Proxy de Adaptação recebe (5), armazena em seu cache, e envia os conteúdos adaptados ao Usuário (6).

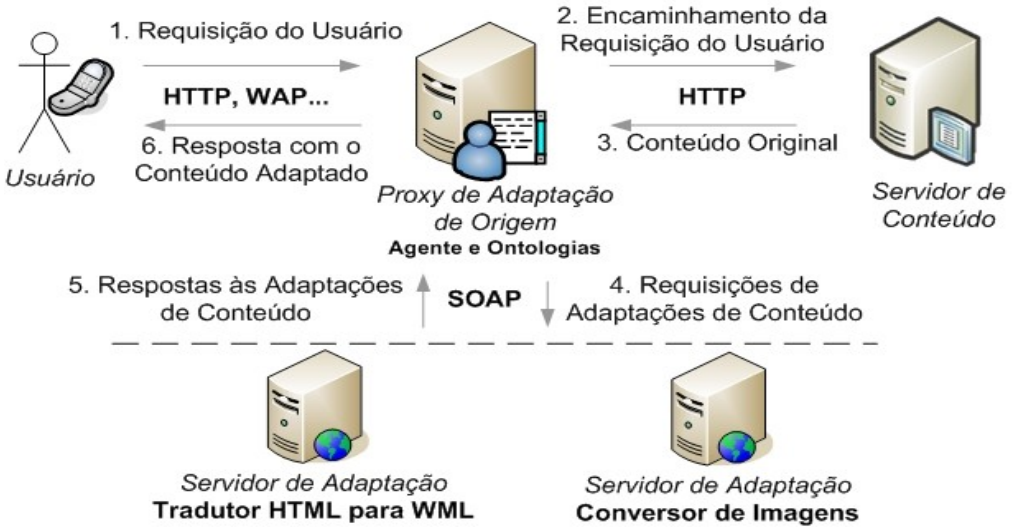

Figura 4. Adaptação Local de Conteúdo.

A Figura 5 considera o caso de uma adaptação remota, quando é necessário um Serviço Web Semântico que não está disponível no Proxy de Adaptação de Origem, nesse exemplo a adaptação de HTML para *Compact HTML* (cHTML). A requisição do usuário é interceptada por um Proxy de Adaptação (1), o agente do Proxy da Adaptação de Origem migra para um Proxy de Adaptação Remoto (2), requisita (3) e recebe (4) o conteúdo a ser adaptado. No Proxy de Adaptação Remoto o Agente utiliza o Servidor de Adaptação necessário (5) e (6), realiza a adaptação e retorna ao Proxy de Adaptação de Origem (7) para enviar o conteúdo adaptado ao Usuário (8).

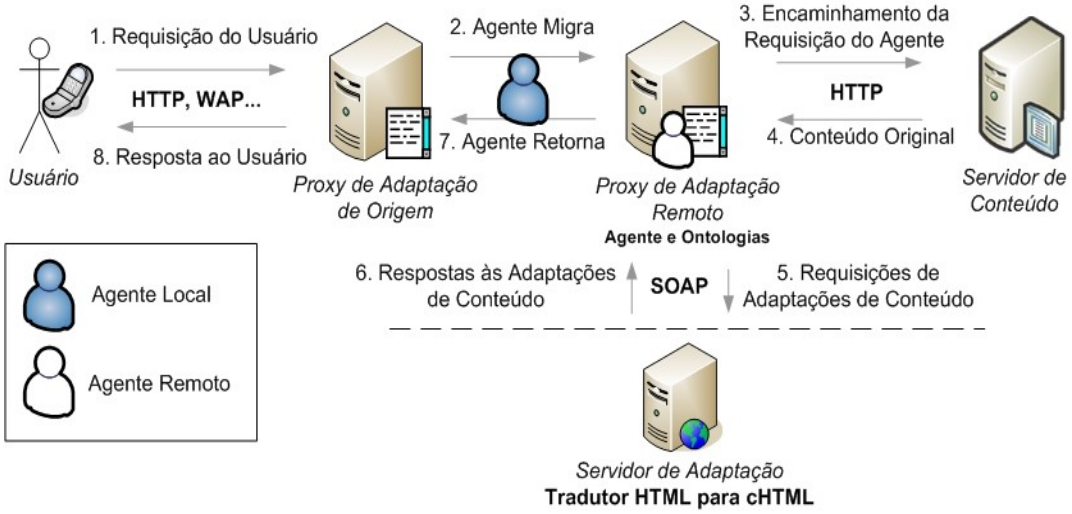

Figura 5. Adaptação Remota de Conteúdo.

5.1. Análise

Dentre os modelos de ontologias especificados tem-se o da Figura 6, que descreve o perfil dos conteúdos adaptados, sendo que *Browser_Accept*, *Others*, *Location*, e *Content_Info* são sub-classes de *Content*. Maiores informações sobre modelos de outros perfis (usuários, redes, dispositivos e contratos de serviços) estão disponíveis em [Forte et al. 2006]. Modelos da UML complementam as especificações nesta etapa, mas não são apresentados por serem bastante conhecidos da comunidade de Engenharia de Software.

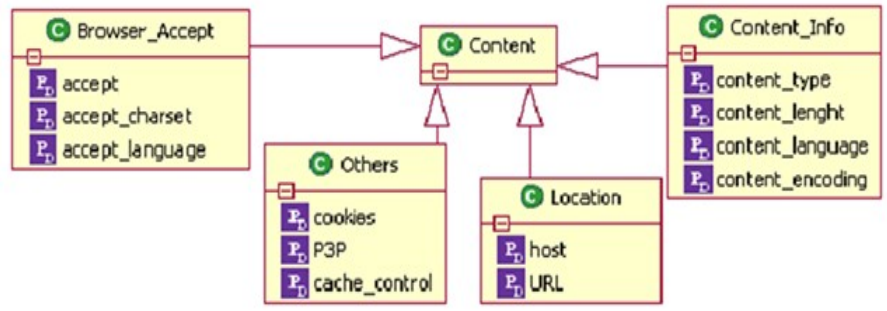

Figura 6. Modelo de ontologia para o perfil de conteúdo.

5.2. Projeto

A partir das Ontologias de domínio, dos modelos UML e dos modelos do FrameAgentesMóveis são obtidos os modelos de Projeto. A Figura 7 apresenta um desses modelos com os componentes reutilizados do framework e os componentes específicos da aplicação (sombreado). O componente *ContentAdaptation* conecta-se com o componente *StrategyChooser* através da interface *IStrategyChooser*, provendo uma estratégia que intercepta as requisições de usuários para adaptações de conteúdo. Esse componente conecta-se com os componentes *ContentTransferProtocol*, *ProfileManager* e *Cache* cujas responsabilidades são: possibilitar a transferência de conteúdos, facilitar a utilização dos perfis, e armazenar conteúdos adaptados para melhorar o desempenho geral da arquitetura.

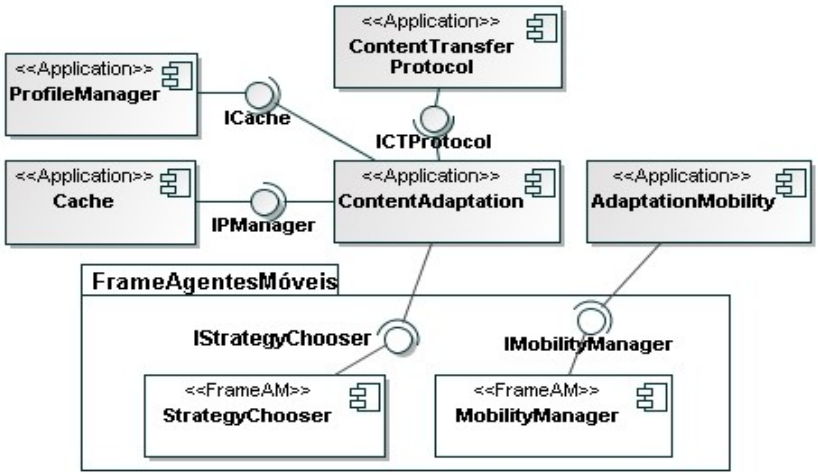

Figura 7. Reutilização do FrameAgentesMóveis.

O componente *AdaptationMobility* conecta-se com o componente *MobilityManager* para definir a mobilidade do agente. Como ilustrado na Figura 8, o fluxo de atividades inicia quando o Proxy de Adaptação de Origem delega uma adaptação de conteúdo ao seu agente. A primeira atividade desse agente é recuperar os perfis que serão utilizados na adaptação. Em seguida, esse agente busca os Serviços Web Semânticos que podem ser utilizados para essa adaptação. Caso nenhum deles seja capaz de realizar sozinho a adaptação, o agente requisita a elaboração um plano de composição de serviços e monitora a execução desse plano, assegurando seu sucesso. Este agente deverá migrar para um Proxy de Adaptação caso seu Proxy de Adaptação de Origem não possua os serviços de adaptação e os perfis necessários ou estiver sobrecarregado.

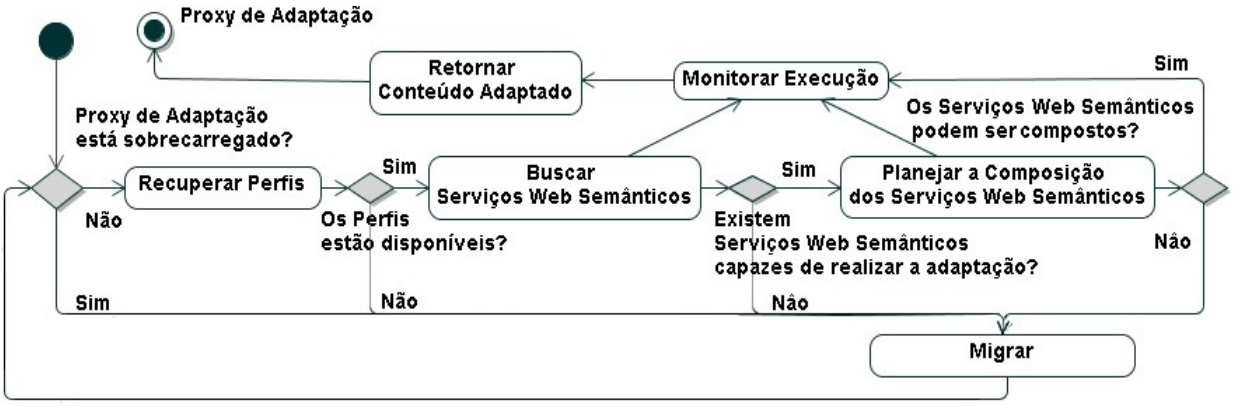

Figura 8. Modelo de Atividades para o Agente Móvel.

Nessa etapa também foram projetados três novos serviços: Tradutor HTML para WML, Conversor de Imagens e Tradutor HTML para cHTML.

5.3. Implementação e Testes

Prosseguindo com a abordagem faz-se a Implementação dos componentes da aplicação (*ProfileManager*, *Cache*, *ContentTransferProtocol*, *ContentAdaptation* e *AdaptationMobility*) e dos Serviços Web Semânticos (*Tradutor HTML para WML*, *Conversor de Imagens* e *Tradutor HTML para cHTML*) previamente projetados. Nessa

etapa foram também adicionadas as descrições OWL-S dos Serviços Web Semânticos, contendo informações, como as entradas, as saídas, e a url de acesso. Por exemplo, na Figura 9 tem-se a descrição do serviço de tradução de HTML para WML.

```
<!-- Service description -->
<service:Service rdf:ID="HTML2WMLService">
      <service:presents rdf:resource="#HTML2WMLProfile"/>
      <service:describedBy rdf:resource="#HTML2WMLProcess"/>
      <service:supports rdf:resource="#HTML2WMLGrounding"/>
</service:Service>

<!-- ServiceProfile description -->
<profile:HTML2WMLService rdf:ID="HTML2WMLrProfile">
      <service:presentedBy rdf:resource="#HTML2WMLService"/>
      <profile:serviceName>HTML2WML</profile:serviceName>
      <profile:hasInput rdf:resource="#htmlpage"/>
      <profile:hasOutput rdf:resource="#wmlpage"/>
</profile:HTML2WMLService >

<!-- ServiceProcess description -->
<process:AtomicProcess rdf:ID="HTML2WMLProcess">...

<!-- ServiceGrounding description -->
<grounding:WsdlGrounding rdf:ID="HTML2WMLGrounding">
 <service:supportedBy rdf:resource="#HTML2WMLService"/>
</grounding:WsdlGrounding>
<grounding:WsdlAtomicProcessGroundingr df:ID="HTML2WMLProcessGrounding">
      <grounding:owlsProcess rdf:resource="HTML2WMLProcess"/>
      <grounding:wsdlDocument>
            http://localhost/HTML2WML/HTML2WMLService?wsdl
      </grounding:wsdlDocument>
```

Figura 9. Descrição de um Serviço Web Semântico.

Finalmente, são realizados os testes da aplicação, conforme ilustra o resultado de uma adaptação de conteúdo apresentado na Figura 10. No caso, uma página da Web, que antes não podia ser apresentada em dispositivos móveis, é adaptada de acordo com as características desse dispositivo.

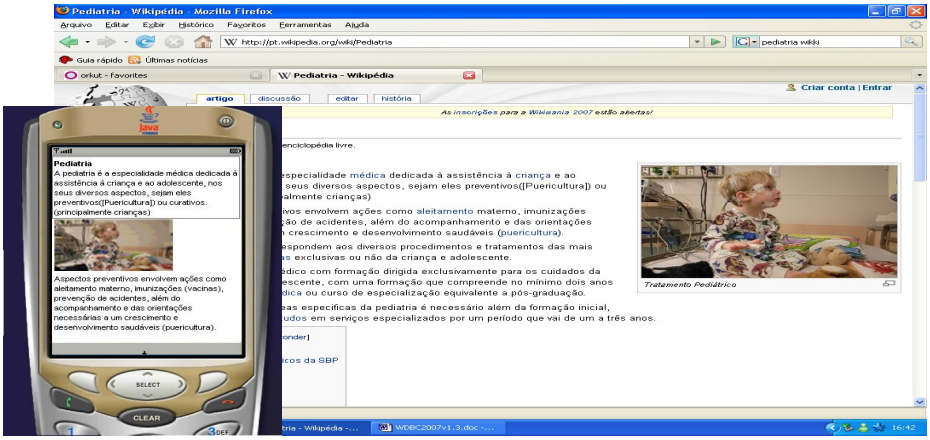

Figura 10. Conteúdo Adaptado.

Concluindo, verificou-se que com as ontologias obteve-se modelos mais genéricos para os componentes da aplicação. Outro ganho foi na implementação onde o tempo foi reduzido devido ao reuso do framework. Além disso, os novos serviços construídos poderão ser reutilizados em outras aplicações do domínio.

6. Trabalhos Correlatos

Existem diversos trabalhos que empregam ontologias, serviços Web e agentes de software para apoiar o Processo de Desenvolvimento de Software. Alguns têm foco apenas nas ontologias como o de [Linhalis e Moreira 2006], que utiliza linguagem natural para descrever os requisitos dos componentes de software, que posteriormente são transformadas em ontologias. A abordagem proposta utiliza os modelos de ontologias complementados por modelos em UML.

Em [Elias et al. 2006] é proposto um repositório de componentes que utiliza descrições sintáticas baseadas em XML para representação de metadados. A abordagem proposta utiliza a linguagem OWL-S que tem semântica para descrever os componentes Serviços Web Semânticos.

O trabalho de [Ha e Lee 2006] usa Serviços Web Semânticos no DBC num ambiente específico de *e-business*. A abordagem proposta é mais genérica e pode ser empregada em outros domínios de aplicações.

Dentre os trabalhos que utilizam Agentes de Software para busca, composição e monitoramento de Serviços Web Semânticos destaca-se o de [Charif-Djebbar e Sabouret 2006]. Este trabalho propõe uma abordagem baseada em sistemas multiagentes, na qual os agentes possuem um protocolo de interação que permite selecionar e compor os serviços. Na abordagem proposta os agentes podem migrar através dos diferentes nós da rede, para utilizar os Serviços Web Semânticos que estão distribuídos.

7. Conclusões e Trabalhos Futuros

Este artigo apresentou uma abordagem para DBC, que emprega Ontologias para descrever domínios e Agentes Móveis para gerenciar Serviços Web Semânticos. Esses agentes migram através dos nós da rede para encontrar os componentes necessários às aplicações, em função dos requisitos do domínio descrito em ontologias e de técnicas da UML. A abordagem dispõe de um framework "caixa-cinza", que possibilita o reuso organizado de componentes e a estruturação das aplicações de acordo com a abordagem proposta. Empregando-se o framework, grande parte da implementação é simplificada.

O estudo de caso ressalta as vantagens do emprego da abordagem proposta, tais como a modificação não antecipada de componentes e o uso de componentes distribuídos. A modificação não antecipada é atendida via a utilização de Serviços Web Semânticos, já que um serviço pode ter seu comportamento alterado sem que seja necessário reconstruir a aplicação. O uso de componentes distribuídos é obtido via a combinação de Serviços Web Semânticos com Agentes Móveis, os quais podem migrar através dos nós da rede para encontrar, compor e monitorar a execução desses serviços.

Dentre os trabalhos futuros destacam-se: (i) refinar a abordagem com base em estudos de casos de outros domínios, como laboratórios virtuais (*WebLabs*) e redes de sensores; (ii) criar ferramentas que auxiliem o Engenheiro de Software nas diferentes

atividades de cada etapa da abordagem; (iii) avaliar o desempenho de aplicações que reutilizam o framework com Serviços Web Semânticos executados por agentes.

Referências

Arango, G. (1989) "Domain analysis – from art to engineering discipline". ACM Sigsoft Software Engineering Notes, vol. 14, no. 3, pp. 152-159.

Balzer, S., Liebig, T. e Wagner, M. (2004) "Pitfalls of OWLS – A Practical Semantic Web Use Case", Anais da International Conference on Service Oriented Computing, pp. 289-298.

Berners-Lee, T., Hendler, J., Lassila, O. (2001) "The Semantic Web" Scientific American, vol. 5, no. 17, pp. 35-43.

Charif-Djebbar, Y. e Sabouret, N. (2006) "Dynamic Service Composition and Selection through an Agent Interaction Protocol". Anais da International Conference on Web Intelligence and Intelligent Agent Technology, pp. 105-108.

Coelho et al. (2007) "Arquitetura e Requisitos de Rede para Web Labs" Anais do Simpósio de Redes de Computadores e Sistemas Distribuídos 2007, pp. 499 – 512.

Ebraert, P., Vandewoude, Y., D'Hondt, T. e Berbers, Y. (2005). "Pitfalls in Unanticipated Dynamic Software Evolution". Anais do Workshop on Reflection, AOP and Meta-Data for Software Evolution, pp. 41-51.

Elias, G., Schuenck, M. Negócio, Y., Dias, J. e Filho, S.M. (2006) "X-ARM: an asset representation model for component repository systems" Anais do Symposium on Applied Computing, pp. 1690 - 1694.

Evermann, J. e Wand, Y. (2005) "Ontology based object-oriented domain modelling: fundamental concepts" Requirements Engineering, vol. 10, no. 5, pp. 146 – 160.

Forte, M., Souza, W.L., e Prado, A.F. (2006) "Utilizando ontologias e serviços web na computação ubíqua". Anais do Simpósio Brasileiro de Engenharia de Software, pp. 287-302.

Dobson, S. et al. (2006) "A Survey of Autonomic Communications" ACM Transactions on Autonomous and Adaptive Systems, vol. 1, no. 2, pp. 223-259.

Gamma E., Helm R., Johnson R. e Vlissides J. (1995) "Design Patterns: Elements of Reusable Object-Oriented Software". Addison-Wesley.

Gruber, T.R. (1993) "A Translation Approach to Portable Ontology Specifications". Knowledge Acquisition, vol.5, no. 2, pp. 199-220.

Ha, Y. e Lee, R. (2006) **"**Integration of Semantic Web Service and Component-Based Development for e-business environment" Anais da International Conference on Software Engineering Research, Management and Applications, pp. 315 – 323.

Hepp, M. (2006) "Semantic Web and Semantic Web Services: Father and Son or Indivisible Twins?" IEEE Internet Computing, vol. 10, no 2, pp. 85–88.

Lee,W. (2007) "Deploying personalized mobile services in an agent-based environment**",** Expert Systems with Applications, vol. 32, no. 4, pp. 1194-1207.

Linhalis, F. e Moreira, D.A. (2006) "Ontology-Based Application Server to the Execution of Imperative Natural Language Requests". Anais do International Conference on Flexible Query Answering Systems", p. 589-600.

Lucrédio, D., Almeida, E.S. e Prado, A.F. (2004) "A Survey on Software Components Search and Retrieval".Anais da Euromicro Conference, pp. 152-159.

OMG. Unified Modeling Language (UML) Specification, Versão 2.1.1, Object Management Group, 2004.

Papazoglou, M.P.(2003) "Service-oriented computing: concepts, characteristics and directions"Anais da Conference on Web Information Systems Engineering, pp. 3–12.

Ross, D. (1977) "Structured Analysis (AS): A Language for Communicating Ideas". IEEE Transactions on Software Engineering, vol. 3, no.1, pp. 16-34.

Santana, L.H.Z. et al. (2007) "Serviço de tradução de linguagens de marcação para a Internet". Anais do XXV Simpósio Brasileiro de Redes de Computadores, vol. 1, pp. 541- 554.

W3C (2004a) OWL Web Ontology Language < http://www.w3.org/TR/owl-features/>

W3C (2004b) OWL-S OWL-S: Semantic Markup for Web Services < http://www.w3.org/Submission/OWL-S/>

Werner, C.M.L. e Braga, R. M.M. (2000) "Desenvolvimento Baseado em Componentes".Anais do Simpósio Brasileiro de Engenharia de Software, pp. 297-329.

Yao, H. e Etzkorn, L. (2004) "Towards A Semantic-based Approach for Software Reusable Component Classification and Retrieval" Anais do 42nd annual Southeast regional conference ACM-SE 42, pp. 110 - 115.