# Utilizando Ontologias e Serviços Web na Computação Ubíqua

**Marcos Forte[+], Wanderley Lopes de Souza, Antonio Francisco do Prado**

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

[+] Centro Universitário Fundação Santo André – Av. Príncipe de Gales, 821
09060-650 - Santo André (SP)

{marcos_forte, desouza, prado}@dc.ufscar.br

***Abstract*** *In the near future, most users will access the Internet by means of small mobile devices. This context of ubiquitous computing is highly volatile because the diversity of device characteristics and access networks extends each day. The flexibility necessary for software reuse in this environment is not provided by the current programming paradigms. Aiming to offer this flexibility this paper proposes the use of ontologies and Web services to extend a framework of components to the domain of content adaptation, which facilitates the development of software based on reuse. A case study illustrates the use of the proposed solution.*

***Resumo*** *Num futuro próximo a maioria dos usuários acessarão a Internet via pequenos dispositivos móveis. Esse contexto da computação ubíqua é altamente volátil, pois a diversidade de características de dispositivos e redes de acesso se amplia a cada dia. A flexibilidade necessária para o reuso de software neste ambiente não é provida pelos atuais paradigmas de programação. Visando oferecer esta flexibilidade este artigo propõe a utilização de ontologias e serviços Web para estender um framework de componentes para o domínio da adaptação de conteúdo, que facilita o desenvolvimento de software baseado no reuso. Um estudo de caso ilustra o uso da solução proposta.*

## 1. Introdução

Pode-se classificar a história da Computação, em função dos ambientes computacionais predominantes, em três eras: a passada dos *mainframes*; a atual dos computadores pessoais; a futura da Computação Ubíqua, denominada por alguns autores de *Pervasive Computing* [Hansmann et al. 2003], que tem por meta permitir ao usuário, usando qualquer dispositivo, a qualquer momento e de qualquer lugar, o fácil acesso e processamento da informação.

A comunicação móvel é um dos fatores que está impulsionando o salto da Computação para essa nova era. A possibilidade de conexão a qualquer momento e de qualquer lugar outorgou aos usuários uma escolha e uma liberdade sem precedentes, permitindo que estes busquem por novas e recompensadoras formas para o tratamento de assuntos pessoais e profissionais. Em apenas uma década as redes móveis permitiram atingir um ritmo de crescimento que demandou quase um século às redes fixas e os

avanços nas tecnologias móveis têm propiciado uma transição de serviços exclusivos de voz para serviços de conteúdo baseados em *Web*.

Esta mobilidade globalizada demanda por novas arquiteturas e protocolos, que permitam a fácil conexão das redes móveis aos diversos tipos de provedores de serviços e de conteúdo espalhados pela Internet. Uma visão futuristica da *Internet Móvel* assume usuários com diferentes perfis utilizando diversos tipos de redes de acesso e uma grande variedade de dispositivos móveis, exigindo serviços personalizados que atendam da melhor forma possível as suas necessidades, disponibilidades e localizações. Nesses contextos da computação ubíqua são descritas informações sobre pessoas, lugares, dispositivos e outros objetos que são considerados relevantes para a interação entre usuários e serviços, incluindo os próprios usuários e serviços.

De forma a facilitar à descrição dessas informações de contexto, a comunidade da *Web* semântica [*Semantic Web Activity*] produziu um conjunto de linguagens e ferramentas para o uso, desenvolvimento, manutenção e compartilhamento de ontologias. Entre as vantagens apresentadas por essas linguagens destacam-se [Knublauch et al. 2006]: são otimizadas para a representação de conhecimento estruturado em um nível elevado de abstração; os modelos de domínio podem ser disponibilizados na Internet e compartilhados entre múltiplas aplicações; é baseada em dialetos não ambíguos da lógica formal, permitindo a utilização de serviços inteligentes baseados no raciocínio (*reasoning*), possibilitando assim, em tempo de execução, a definição dinâmica de classes, a re-classificação de instâncias e a execução de consultas lógicas complexas. Além disso, essas linguagens operam sobre estruturas similares às linguagens orientadas a objeto, e podem ser eficazmente integradas aos componentes de software tradicionais.

Dentre os diferentes domínios da computação ubíqua destaca-se o da adaptação de conteúdo para dispositivos móveis, que é o enfoque desta pesquisa. A adaptação de conteúdo consiste em adaptar conteúdo e aplicações de modo automático, respeitando as preferências dos usuários e as características do contexto da computação ubíqua. Como existe uma infinidade de adaptações possíveis, quanto maior a quantidade de serviços de adaptação disponíveis, maior a chance de satisfazer as necessidades dos usuários. Neste caso o uso de serviços *Web* e seus padrões facilitam o desenvolvimento de novos servidores, pois permitem independência de linguagens e plataformas.

Prover soluções adaptáveis a esses contextos tão diversos, que facilite o desenvolvimento de aplicações baseado no reuso, constitui um grande desafio para a Engenharia de Software. Assim, motivados por estas idéias, este artigo propõe a utilização de ontologias e serviços *Web* no contexto da computação ubíqua para facilitar o desenvolvimento e incentivar o reuso de software. Ontologias e serviços *Web* foram usados para reconstruir um *framework*, denominado FACI (***F****ramework* para **A**daptação de **C**onteúdo na **I**nternet) [Claudino, Souza e Prado 2005], com o propósito de apoiar o desenvolvimento de aplicações no domínio da adaptação de conteúdo, baseado no reuso da modelagem e do código. A reconstrução do FACI resultou na adaptação e adição de novos componentes mais genéricos, o que ampliou o seu reuso para diferentes aplicações desse domínio de adaptação de conteúdo. A seqüência deste artigo está estruturada da seguinte forma: a seção 2 versa sobre o tema ontologias e sobre algumas linguagens, desenvolvidas para a especificação das mesmas, e trata também da integração com serviços *Web*; a seção 3 concentra-se no uso de ontologias e serviços

*Web* na computação ubíqua; a seção 4 apresenta a reconstrução do FACI e um estudo de caso baseado no reuso dos componentes dessa nova versão; a seção 5 descreve os trabalhos correlatos; e finalmente a seção 6 apresenta as conclusões.

## 2. Ontologias

Para alcançar a interoperabilidade entre sistemas heterogêneos, envolvidos na execução de um determinado domínio de aplicações, é fundamental que estes possam compartilhar informações de maneira automática, com um entendimento comum e não ambíguo dos termos e conceitos usados nessas aplicações. Neste sentido as ontologias constituem-se em artefatos importantes na viabilização do tratamento dessa heterogeneidade.

Berners-Lee propôs a *Web*-Semântica [Berners-Lee, Hendler e Lassila 2001] como uma evolução natural da *Web* vigente, a fim de viabilizar a manipulação do conteúdo por parte das aplicações com capacidade de interpretar a semântica das informações. O conteúdo da *Web*, que não passa de informação sem significado para os computadores, pode então ser interpretado por máquinas através do uso de ontologias, tornando a recuperação da informação na *Web* menos ambígua e fornecendo respostas mais precisas às consultas dos usuários.

Ontologias são usadas para diversos fins, variando do apoio aos processos de desenvolvimento de software, ou qualquer outra atividade executada por equipes geograficamente distribuídas, ao apoio em tempo de execução aos sistemas de informação [Guarino 1998]. Neste artigo o termo ontologia refere-se a uma descrição explícita de conceitos e relações de um domínio de aplicação, incluindo um vocabulário de termos empregado nesse domínio e um conjunto de axiomas que expressam as restrições para a interpretação desse vocabulário.

Para a construção de ontologias é essencial dispor-se de uma linguagem apropriada. Essa linguagem deve possuir uma semântica bem-definida, ser expressiva o suficiente para descrever inter-relacionamentos complexos e restrições entre os objetos, ser capaz de manipular e inferir automaticamente, isso tudo dentro de limites aceitáveis de tempo e recursos [Martin e Mcllraith 2003].

O *World Wide Web Consortium* (*W3C*) orienta o desenvolvimento, a organização e a padronização de linguagens para promover a interoperabilidade entre aplicações na Web. Dentre essas linguagens destacam-se: *Resource Description Framework (RDF)* [Monola e Miller 1999] e mais recentemente a *Web Ontology Language (OWL)* [McGuinness e Harmelen 2004].

### 2.1. *Web* Ontology Language (*OWL*)

A *OWL* é uma linguagem de marcação semântica usada na publicação e compartilhamento de ontologias na *Web*. Na *OWL* ontologia é um conjunto de definições de classes, propriedades e restrições em relação aos modos com que essas classes e propriedades podem ser empregadas.

Uma ontologia *OWL* pode incluir: relações de taxonomia entre classes; propriedades dos tipos de dados e descrições dos atributos de elementos das classes; propriedades do objeto e descrições das relações entre elementos das classes; instâncias das classes; e instâncias das propriedades. As propriedades de tipos de

dados e as propriedades de objeto são coletivamente as propriedades de uma classe.

Uma das vantagens na utilização da *OWL* é sua base em um dialeto não ambíguo da lógica formal chamado *Descrição Lógica* [Baader et al. 2003]. O suporte formal torna possível a utilização de serviços inteligentes baseados em raciocínio, tais como a classificação e verificação automática de consistência. Esses serviços podem ser usados em tempo de criação, de modo a facilitar a construção de modelos reusáveis e bem-testados do domínio, ou em tempo de execução, possibilitando a definição dinâmica de classes, re-classificação de instâncias e execução de consultas lógicas complexas.

### 2.2. Serviços *Web* Semânticos

Os serviços *Web*, que há anos são a base das arquiteturas orientadas a serviços, começam a apresentar deficiências principalmente nas áreas de descrição, descoberta e composição de serviços. O problema está na falta de suporte semântico na linguagem de descrição de serviços (*Web Services Description Language - WSDL*) e no mecanismo de armazenamento e pesquisa de serviços (*Universal Description, Discovery and Integration - UDDI*).

Com o propósito de integrar a *Web* semântica aos serviços *Web*, foram desenvolvidas novas linguagens para descrição e composição de serviços apresentadas na Figura 1. Essa integração ainda está em curso, e as atuais extensões semânticas do *UDDI* [Naumenko et al. 2005],[Srínívasan, Paoluccí e Sycara 2005] não garantem a compatibilidade e performance necessárias.

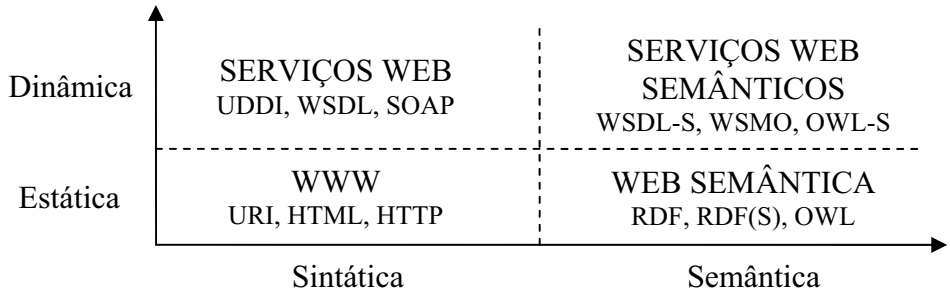


**Figura 1. Evolução das tecnologias *Web***

#### 2.2.1. *Web Ontology Language for Services* (*OWL-S*)

*OWL-S* [Martin 2004] é uma linguagem de descrição de serviços *Web*, que une a *WSDL* [WSDL] com a informação semântica do *OWL*, sendo organizada em três módulos: *Profile*; *Process Model* e *Grounding*. O *Profile* descreve as capacidades e características adicionais de serviços *Web*, com base nas transformações que esses serviços produzem, auxiliando na decisão do requisitante quando este utiliza um determinado serviço *Web*.

O *Process Model* especifica o protocolo de interação, que possibilita ao requisitante saber, num dado momento da transação, que informação enviar e receber do fornecedor. Esse módulo distingue dois tipos de processos: atômicos e compostos. Os atômicos fornecem descrições abstratas das informações trocadas com os requisitantes, correspondendo às operações que o fornecedor pode executar diretamente. Os

compostos são usados para descrever coleções de processos (atômicos ou compostos), organizados através de algum tipo de estrutura de controle de fluxo.

O *Grounding* descreve como processos atômicos são transformados em mensagens concretas, de modo que possam ser trocados via rede ou através de uma chamada de procedimento. É definido como um mapeamento "um para um" de processos atômicos para a especificação de descrições *WSDL*.

## 3. Uso de Ontologias e Serviços *Web* na Computação Ubíqua

Baseado nos estudos apresentados pesquisou-se o uso de ontologias para descrever características do contexto da computação ubíqua, concentrando-se na solução de problemas relacionados com a adaptação de conteúdo em dispositivos móveis, tais como celulares e *palmtops*.

Informações sobre as características do contexto são elementos fundamentais no desenvolvimento de soluções para a adaptação de conteúdo e compõe a política de adaptação. Essa política decide quais serviços de adaptação serão oferecidos, quais adaptadores locais ou remotos executarão essas adaptações e quando estas deverão ser solicitadas.

Visando aperfeiçoar o processo de tomada de decisão, a política de adaptação considera ainda as regras de adaptação, que indicam as condições que devem ser atendidas para que as ações correspondentes a uma determinada adaptação sejam executadas.

Finalmente, é necessário disponibilizar uma infra-estrutura de servidores de adaptação distribuídos pela Internet. O uso da tecnologia de serviços *Web* oferece uma solução consistente, com uma grande quantidade de ferramentas e padrões bem definidos. Além disso, o uso de ontologias e a incorporação de semântica em seus padrões possibilitam uma fácil migração das atuais soluções proprietárias, de descoberta e composição de serviços, para uma arquitetura distribuída baseada na *Web* Semântica.

### 3.1. Perfis

Para descrever o contexto da computação ubíqua, visando à eficácia da política de adaptação, as seguintes informações são necessárias: características e capacidades dos dispositivos de acesso; dados pessoais e preferências dos usuários; condições da rede de comunicação; características dos conteúdos requisitados; resoluções contratuais entre o provedor de serviços e o usuário final. Essas informações estão contidas, respectivamente, nos seguintes perfis: *dispositivo*, *usuário*, *rede*, *conteúdo* e *Service Level Agreement (SLA).* Esses perfis são descritos em *OWL*.

Também são necessárias informações sobre os serviços de adaptação disponíveis. Neste caso, além das características dos serviços oferecidos, devem ser incluídas informações sobre comunicação (e.g., protocolos, endereçamento) e as condições para a execução de determinada adaptação. Essas condições são baseadas em regras e utilizadas pela política de adaptação. Os perfis de serviço são descritos em *OWL-S.*

### 3.1.1. Perfis de dispositivo, usuário, rede, conteúdo e *SLA*

O perfil de dispositivo contém características de hardware e software do dispositivo. O conhecimento das capacidades de um dispositivo orienta o processo de adaptação, a fim de que apenas as adaptações necessárias sejam aplicadas ao conteúdo (e.g., remoção de som de um conteúdo requisitado por um dispositivo incapaz de reproduzir som). O campo *UserAgent*, do cabeçalho *HTTP*, é empregado para determinar o dispositivo de acesso utilizado pelo usuário, permitindo assim consultar o perfil desse dispositivo que está armazenado na base de dados. A Figura 2 apresenta um dos modelos de ontologia, baseado no *EMF Ontology Definition Metamodel* (*EODM*) [EODM 2006], para a descrição de perfis de dispositivo. Para que se tenha uma melhor idéia das vantagens do uso de ontologias para a descrição de perfis, o modelo permite a inserção de restrições que auxiliam na consistência e validação do perfil. Por exemplo, a restrição *Supported_ImageRestriction* define que a classe *Supported_Image* só possa ser instanciada com os indivíduos declarados na classe enumerada *Image_Format*.

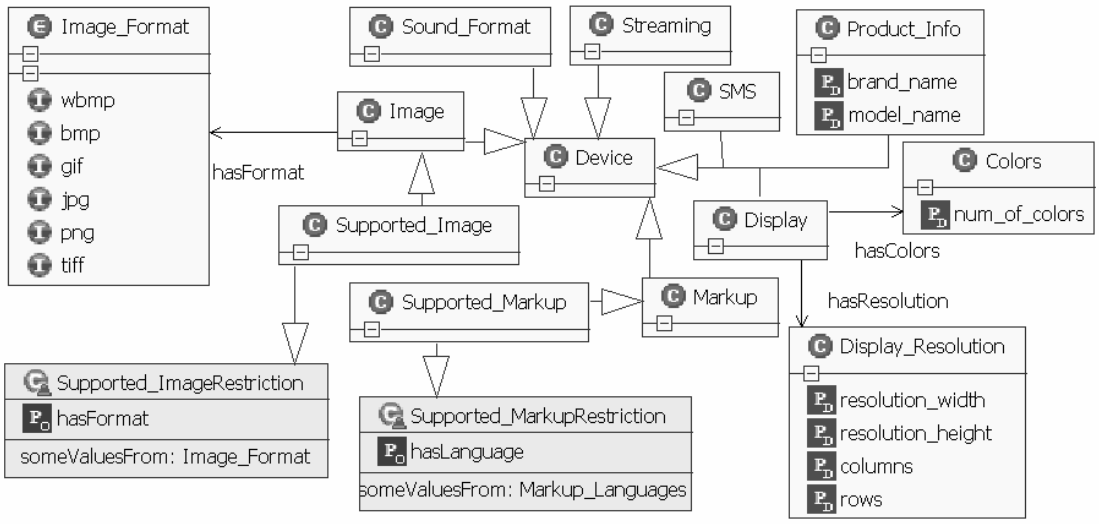


**Figura 2. Modelo de ontologia para a descrição de perfis de dispositivo.**

Atualmente os padrões de perfis de dispositivos móveis mais adotados pelos fabricantes são o *User Agent Profile (UAPROF)* [OMA 2003] e o *Composite Capability /Preference Profiles (CC/PP)* [Klyne et al. 2004], sendo que ambos deveriam produzir especificações equivalentes, já que o *UAPROF* seria um vocabulário específico com base na estrutura do *CC/PP* [Cannistrà 2003]. Infelizmente os desenvolvedores do *UAPROF* acabaram criando um novo esquema (www.openmobilealliance.org/tech/ profiles/UAPROF/ccppschema-20030226), que redefiniu alguns elementos do *CC/PP* (www.w3.org/2002/11/08-ccpp-schema). Por esse motivo a equivalência lógica entre elementos das diferentes especificações não pode ser reconhecida em termos de *RDF*. Para resolver esse problema, o uso de ontologias, baseadas em *OWL*, permite o mapeamento dos elementos de diferentes padrões, possibilitando o reuso de perfis que utilizem essas especificações.

O perfil de usuário contém informações pessoais deste e de suas preferências de adaptação de conteúdo. Diferentes usuários podem desejar que diferentes adaptações sejam aplicadas a um conteúdo requerido (e.g., enquanto um usuário tem preferência pela redução da resolução de imagens, um outro pode preferir pela redução do número

de cores). Adaptações, não baseadas nas preferências dos usuários, podem ser inconvenientes ou até mesmo indesejadas.

A Figura 3 apresenta um dos modelos de ontologia para a descrição de perfis de usuário. Esse modelo representa a classe *USER* com as propriedades *ID*, que é usada para recuperar as informações do usuário na base de dados, e *Name,* que identifica o nome do usuário. Nas sub-classes de *Services* estão descritas as propriedades dos serviços de adaptação, que podem ser configuradas pelo usuário de acordo com as suas preferências. Para exemplificar, na classe *Classification_and_Filtering* há a propriedade *Filter_Profile*, onde são definidos quais tipos de conteúdo serão bloqueados (e.g., sexo, shopping, jogos, violência).

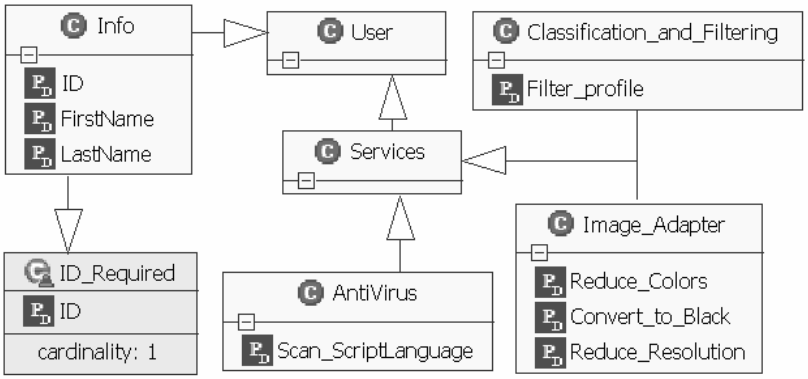


**Figura 3. Modelo de ontologia para descrição de perfis de usuário.**

O perfil de rede é obtido dinamicamente através de agentes, que monitoram parâmetros da rede de comunicação entre o provedor e o usuário. As informações contidas nesse perfil orientam alguns processos de adaptação (e.g., imagens, vídeo e áudio sob demanda), a fim de que o conteúdo adaptado esteja otimizado para as condições da rede num determinado momento.

O perfil de conteúdo também é obtido dinamicamente, sendo baseado em características do próprio conteúdo requisitado. A partir de informações extraídas do cabeçalho *HTTP* (e.g., se o conteúdo dispõe de texto e/ou imagem, idioma) e do conjunto de meta dados do conteúdo, se disponível, são determinadas as alterações necessárias e aplicáveis ao conteúdo.

No perfil *SLA* estão descritas as resoluções contratuais entre o provedor de acesso e o usuário. Atualmente esses provedores oferecem diferentes planos aos seus usuários, incluindo largura de banda, tempo de conexão e vários serviços de valor agregado, permitindo assim que os usuários escolham o plano mais adequado as suas necessidades.

### 3.1.2. Perfis de Serviços

O perfil de serviço descreve as características de um serviço de adaptação, incluindo as condições necessárias para a sua execução. Essas características podem ser divididas em duas classes: as diretamente associadas ao processo de adaptação, representadas pelas Entradas e Saídas; e as de apoio, representadas pelas Pré-condições e Efeitos, que auxiliam a decisão da política de adaptação em executar, ou não, determinado serviço, conforme ilustra a Figura 4.

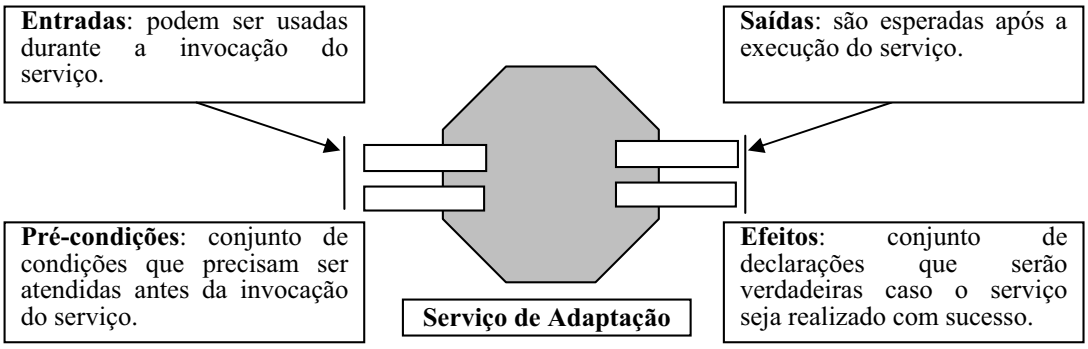


**Figura 4. Características de perfis de serviço.**

Existem outras funções que destacam esse perfil em relação aos outros, e que determinaram o uso do *OWL-S* para sua descrição. A principal delas, é que nesse perfil é realizado o mapeamento das descrições semânticas (*OWL*) dos perfis, com as descrições sintáticas (*WSDL*) dos serviços *Web,* permitindo a integração de ontologias com serviços *Web.* Além disso, ele também descreve as características de comunicação (e.g., protocolos, endereços) dos serviços *Web*, facilitando o acesso remoto a esses serviços.

### 3.2. Descoberta e Composição de Serviços

A partir da combinação das informações do ambiente ubíquo, descritas nos perfis de *dispositivo*, *usuário*, *rede*, *conteúdo* e *SLA*, e das informações dos perfis de serviço, são definidos os serviços necessários para a execução de uma determinada adaptação. Caso seja necessário mais de um serviço para realizar a adaptação necessária, deve-se gerenciar a ordem de suas execuções.

Para auxiliar na determinação dos serviços necessários, os perfis de serviço podem conter regras especificadas em *Semantic Web Rule Language (SWRL)* [Horrocks 2003]. Essa abordagem permite que os componentes responsáveis pela política de adaptação, sejam independentes das variações provocadas pelas mudanças e inclusões de novos serviços e regras.

Com o propósito de exemplificar a especificação, baseada em *OWL-S*, dos perfis de serviço, é apresentado na Figura 5 o esqueleto do perfil de um serviço de conversão de linguagens de marcação. Inicialmente estão descritas as ontologias que serão importadas, de modo a incluir as informações semânticas já definidas (1). Nesse caso são reutilizados: uma ontologia pública, *semwebglossary.owl,* que define termos da *Web* semântica com o propósito de evitar ambigüidade de definições; os perfis de conteúdo (*content.owl*) e dispositivo (*device.owl*). Na seqüência é definida uma classe, que descreve as linguagens de marcação suportadas (2), e uma propriedade *canBeConvertedTo* (3), que será utilizada na definição das possíveis conversões realizadas pelo serviço (4). Em (5) é declarado o parâmetro que descreve a linguagem de marcação a ser convertida. Essa informação é obtida da classe *Content_Type* do perfil de conteúdo, que é previamente instanciada com informações sobre o conteúdo requisitado. Do mesmo modo, a linguagem de marcação desejada é obtida através do parâmetro, definido na classe Supported*_Markup* do perfil de dispositivo, que informa as linguagens suportadas (6). As entradas, as saídas, as precondições e os efeitos estão definidos na descrição do processo (7). A pré-condição baseia-se em uma regra em

SWRL (8), que retorna verdadeiro quando a conversão necessária está entre as definidas em (4). A seção de *Grounding*, entre outras que não estão apresentadas neste exemplo por limitações de espaço, é gerada automaticamente a partir da descrição *WSDL* do serviço.

```
...
<!ENTITY gloss "http://www.personal-reader.de/rdf/semwebglossary.owl">
<!ENTITY device "http://www.adaptationsrv.org/device.owl">
<!ENTITY content "http://www.adaptationsrv.org/content.owl">
...
<owl:Class rdf:ID="SupportedMarkupLanguage">
  <owl:oneOf rdf:parseType="Collection">
     <factbook:Language rdf:about="&glossary;#HTML"/>
     <factbook:Language rdf:about="&glossary;#XHTML"/>
     <factbook:Language rdf:about="&glossary;#XML"/>
     <factbook:Language rdf:about="&glossary;#CHTML"/>
     <factbook:Language rdf:about="&glossary;#WML"/>
  </owl:oneOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="canBeConvertedTo">
  <rdfs:domain rdf:resource="#SupportedMarkupLanguage"/>
  <rdfs:range rdf:resource="#SupportedMarkupLanguage"/>
</owl:ObjectProperty>
<rdf:Description rdf:about="&glos;#HTML"><canBeConvertedTo rdf:resource="&glos;#WML"/></rdf:Description>
<rdf:Description rdf:about="&glos;#HTML"><canBeConvertedTo rdf:resource="&glos;#XHTML"/></rdf:Description>
<rdf:Description rdf:about="&glos;#HTML"><canBeConvertedTo rdf:resource="&glos;#XML"/></rdf:Description>
<rdf:Description rdf:about="&glos;#HTML"><canBeConvertedTo rdf:resource="&glos;#cHTML"/></rdf:Description>
<rdf:Description rdf:about="&glos;#XML"><canBeConvertedTo rdf:resource="&glos;#XHTML"/></rdf:Description>
...
<process:Input rdf:ID="InputMarkupLanguage">
 <process:parameterType rdf:datatype="&xsd;#anyURI">&content;#Content_Type
 </process:parameterType>
</process:Input>
<process:Input rdf:ID="OutputMarkupLanguage">
 <process:parameterType rdf:datatype="&xsd;#anyURI">&device;#Supported_Markup
 </process:parameterType>
</process:Input>
...
<process:AtomicProcess rdf:ID="MarkupConverterProcess">
 <process:hasInput rdf:resource="#InputMarkupLanguage"/>
 <process:hasInput rdf:resource="#OutputMarkupLanguage"/>
 <process:hasInput rdf:resource="#InputString"/>
 <process:hasOutput rdf:resource="#OutputString"/>
 <process:hasPrecondition rdf:resource="#SupportedConvertion"/>
 <process:hasEffect rdf:resource= ="#MarkupLanguageConverted”/>
</process:AtomicProcess>
...
<expr:SWRL-Condition rdf:ID="SupportedConvertion">
  <rdfs:label>canBeConvertedTo(InputMarkupLanguage, OutputMarkupLanguage)</rdfs:label>
  <expr:expressionLanguage rdf:resource="&expr;#SWRL"/>
  <expr:expressionObject><swrl:AtomList><rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#canBeConvertedTo"/>
          <swrl:argument1 rdf:resource="#InputMarkupLanguage"/>
          <swrl:argument2 rdf:resource="#OutputMarkupLanguage"/>
        </swrl:IndividualPropertyAtom>
     </rdf:first></swrl:AtomList></expr:expressionObject>
</expr:SWRL-Condition>
...
```

**Figura 5. Esqueleto do perfil de serviço de conversão de linguagens de marcação.**

Algumas vezes não é possível encontrar um único serviço que satisfaça todos os objetivos necessários. Uma solução para este problema pode ser obtida a partir da composição de serviços, combinados da maneira apropriada para atender as metas pretendidas.

Inicialmente é necessário identificar os serviços *Web*, que farão parte da composição, assumindo que os mesmos são definidos pelos seguintes elementos: Entradas *I*, Saídas *O*, Pré-Condições *P* e Efeitos *E*. Seja S = {$S_1$, $S_2$,..., $S_n$} um conjunto de serviços, onde cada $S_i$ é definida por uma quádrupla $\left(I_s^i, O_s^i, P_s^i, E_s^i\right)$, e seja M = {$M_1$, $M_2$,...,$M_n$} um conjunto de metas, onde cada meta $M_j$ é definida por uma quádrupla $\left(I_m^j, O_m^j, P_m^j, E_m^j\right)$. Se existe um $S_i$ tal que:

1) $\bigcup_{i=1}^{m} I_s^i \subseteq \bigcup_{j=1}^{n} I_m^n$ e $\bigcup_{i=1}^{m} P_s^i \subseteq \bigcup_{j=1}^{n} P_m^n$ ; e,

2) $\bigcup_{i=1}^{m} O_s^i \supseteq \bigcup_{j=1}^{n} O_m^n$ e $\bigcup_{i=1}^{m} E_s^i \supseteq \bigcup_{j=1}^{n} E_m^n$

são incluídos na lista de possíveis serviços a serem aplicados na adaptação de um conteúdo. Caso somente a primeira parte seja satisfeita, significando que não foi encontrado um serviço que realize sozinho a adaptação necessária, é ativado o algoritmo de composição. Uma das técnicas utilizadas para a composição de serviços é a *forward chaining* [Lara et al. 2005]. Essa técnica baseia-se na seleção de um possível serviço *S*, que contenha as entradas requeridas por *O* e verifica se os elementos de saída são satisfeitos. Caso *S* não atinja o objetivo, um novo objetivo *O'* será definido a partir de *O* e das saídas e efeitos geradas por *S* e todo o processo se repete.

## 4. Extensão do FACI

Para o domínio de aplicações alvo desta pesquisa, os casos de uso foram modelados e implementados no *Framework para Adaptação de Conteúdo na Internet (FACI)* [Claudino, Souza e Prado 2005], estendido de modo a suportar ontologias e serviços Web, aproveitando a sua estrutura básica para o desenvolvimento de diferentes aplicações de adaptação de conteúdo. Esse *framework* foi organizado em dois pacotes principais: *Adaptation Proxy*, que desempenha o papel do dispositivo de borda (e.g., proxy); e *Adaptation Server*, que desempenha o papel do módulo de serviços de adaptação. No *FACI,* os atores, usuário da Internet (*User*) e servidor de origem (*Origin Server*) interagem com o *Adaptation Proxy* através das requisições de conteúdo e suas respostas. O *Adaptation Proxy* oferece recursos para analisar essas requisições e respostas, implementa a política de adaptação e pode realizar adaptações localmente ou solicitá-las aos *Adaptation Servers*, que as realizarão remotamente.

No nível de componente a preocupação é com a estrutura interna do sistema para atender as suas funcionalidades. Para tal o *FACI* foi estruturado num conjunto de componentes combinados, que disponibilizam seus serviços através de suas interfaces. A Figura 6 apresenta os componentes do FACI, reconstruído com base em ontologias e serviços *Web*. No modelo estão ressaltados os componentes adaptados e adicionados com a extensão do FACI.

O *Callout Protocol Clien*t, que é responsável pela comunicação com o *Adaptation Server*, foi adaptado com a adição do *Simple Object Access Protocol (SOAP)* [SOAP], proporcionando a compatibilidade com serviços *Web*. O *Internet Content Adaptation Protocol (ICAP)* [Elson e Cerpa 2003] foi mantido de modo a suportar servidores já desenvolvidos. O perfil de serviço descreve qual protocolo será utilizado para a comunicação com o respectivo serviço.

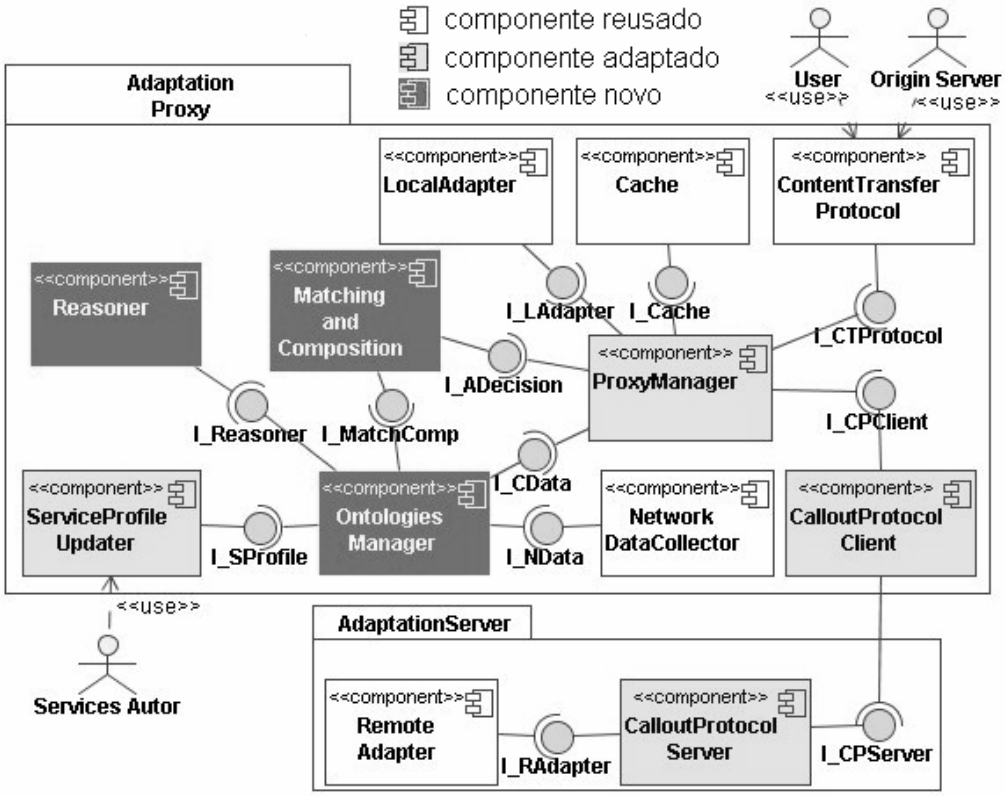


**Figura 6. Componentes do *Framework para Adaptação de Conteúdo na Internet*.**

*Ontologies Manager*, um novo componente que foi introduzido para gerenciar o armazenamento, a recuperação e o processamento das ontologias estáticas (perfis de dispositivo, usuário, *SLA* e serviços) numa base de dados relacional. Esse componente também trata os perfis de serviços, inseridos através do *Service Profile Updater*. Este último, que foi adaptado para realizar as funções descritas, disponibiliza uma interface Web para a inserção dos perfis de serviços, onde o provedor de serviço deverá inserir informações sobre os serviços de adaptação oferecidos, que serão convertidos para a especificação OWL-S. Para possibilitar o suporte a serviços *Web*, o *Service Profile Updater* converte as descrições de serviço *Web*, baseadas em *WSDL*, em um módulo *Grounding* do *OWL-S*. As ontologias dinâmicas, armazenadas em memória, são geradas sob demanda a partir das informações obtidas pela interface *I_NData*, para a geração do perfil de rede, e *I_CData*, para a geração do perfil de conteúdo. A interface I_CData foi adicionada a partir da adaptação do componente ProxyManager.

*Matching and Composition*, outro componente adicionado, realiza a função da política de adaptação no *Framework* a partir do cruzamento das informações do contexto de computação ubíqua com as características dos perfis de serviços. Para realizar essa tarefa são realizadas consultas baseadas em *SPARQL* [Prud'hommeaux e Seaborne 2006], utilizando além das informações declaradas nos perfis, as inferidas pelo *Reasoner,* para localizar os serviços que possuam entradas e saídas compatíveis com o contexto. Os resultados obtidos passam por uma segunda filtragem que verifica a conformidade com as pré-condições e os efeitos. Caso o resultado final aponte a necessidade da composição de serviços, esse componente também gerenciará a ordem de execução dos serviços de adaptação. Através da interface *I_ADecision* são enviadas

as informações, sobre o serviço de adaptação a ser invocado, ao *Proxy Manager* que irá, através do *Callout Protocol Client,* requisitar o serviço ao respectivo *Adaptation Server***.** A invocação de serviços é executada recursivamente até que o último serviço da composição seja realizado, liberando o *Proxy Manager* para o envio da resposta adaptada ao *User.*

No pacote *Adaptation Server*, o componente *Callout Protocol Server* foi adaptado, para suportar a tecnologia de serviços *Web, com a adição do SOAP*. Porém, com as alterações feitas no *Adaptation Proxy*, qualquer servidor de aplicações compatível com os padrões de serviços *Web* pode assumir a função de Adaptation Server.

Para que se tenha uma melhor compreensão do emprego de ontologias e serviços *Web* na política de adaptação, a Figura 7 apresenta um contexto de adaptação com perfis de uma aplicação, utilizando a notação perfil.classe, onde um usuário acessa a *Web* via celular. Quando o usuário conecta-se a *Web*, o provedor de acesso envia seu *User_ID*, conjuntamente com a requisição, para o *Adaptation Proxy*. A partir do *User_ID* o *Ontologies Manager* busca, na sua base de dados, os perfis de usuário e *SLA*. Da requisição do usuário é extraído o *User Agent (UA)*, que identifica o modelo do dispositivo de acesso, possibilitando assim a busca do perfil de dispositivo. O perfil de rede é criado dinamicamente a partir das informações obtidas pelo *Network Data Collector* e o perfil de conteúdo é gerado a partir de informações extraídas dos campos do cabeçalho da mensagem de resposta. Neste exemplo o usuário contratou o serviço pago de classificação e filtragem de conteúdo, tendo a propriedade *Pay_for_Filtering*, do perfil *SLA*, configurada como *true*.

| | |
|---|---|
| **User**.Filter_Profile – 001 | **Network**.WAN – GPRS |
| **SLA**.Services – Pay_for_Filtering | **Network**.RTT – 10 |
| **Device**.DisplayResolution – 320x200 | **Content**.Type – HTML |
| **Device**.Supported_Markup – XHTML | **Content**.URL – www.test.com |

**Figura 7. Exemplo de um contexto de adaptação.**

Uma vez coletadas as informações dos perfis, é definido o objetivo, *O = (HTML, XHTML, Pay_ for_Filtering*, _ ), e são iniciadas as buscas de serviços para a realização das adaptações necessárias. O componente *Matching and Composition* descobre que o serviço de classificação e filtragem de conteúdo, $S_x$ *= (HTML, HTML, Pay_for_Filtering*, _ ), atende às entradas e pré-condições, mas não suporta a saída em *XHTML*. Como todos objetivos pretendidos não foram atingidos, é criado um novo objetivo, *O' = (HTML, XHTML*, _ , _ ), sendo que em uma nova busca o serviço de tradução de linguagem de marcação, $S_y$ *= (HTML, XHTML*, _ , _ ), atendeu aos requisitos e a composição é concluída.

A extensão do FACI foi implementada em Java, tendo sido utilizadas várias *APIs* disponíveis publicamente. No componente *Ontologies Manager* foram usadas a *API OWL-S* [*OWL-S*], que estende a *API JENA* [*Jena*], e o banco de dados relacional *MySQL*. No *Reasoner* foi usada a *API Pellet* [*Pellet*] e no *Matching and Composition* foi usado o módulo *SPARQL* do *API OWL-S.*

### 4.1. Estudo de Caso

Para testar a performance e a capacidade de carga do FACI com suporte a ontologias e serviços *Web*, foi realizado um estudo de caso envolvendo dois computadores. Um com sistema operacional *Linux Fedora Core 2* (2Ghz – 256MB) e o outro com *Windows XP* (3Ghz – 1GB). Para evitar que alterações no tempo de resposta dos servidores de origem (incluindo variações no *throughput* da Internet) afetassem o resultado final, um servidor *Apache 2* foi instalado, empregando-se o recurso *hosts virtuais*. Isto permitiu a clonagem das páginas testadas, restringindo o fluxo de dados aos computadores envolvidos no estudo de caso. No *Linux* também foi instalado um servidor *Tomcat/Axis*, com os respectivos *remote adapters*, que exerceu o papel de *adaptation server*. Na plataforma *Windows XP* foram instalados o *adaptation proxy* com os perfis de serviços de classificação e filtragem de conteúdo e de conversão de linguagens de marcação.

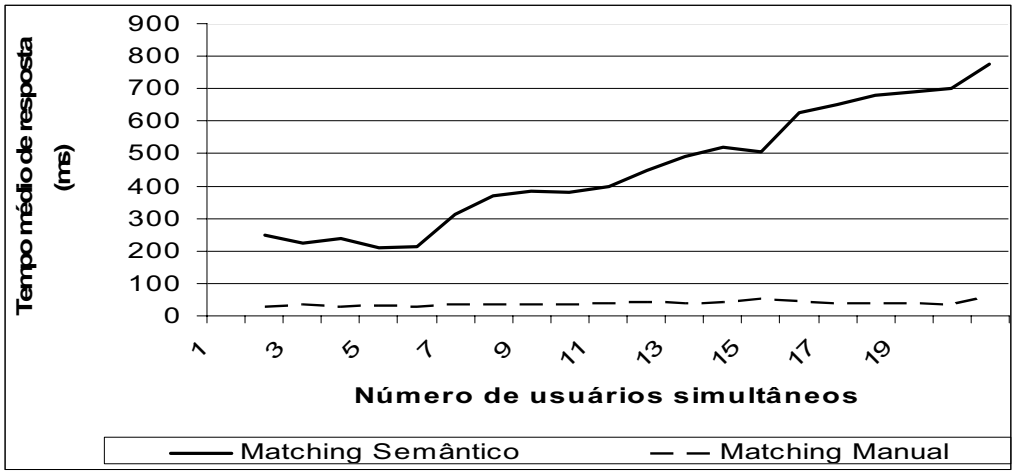


Figura 8. Tempo Médio de Resposta x Número de Usuários

A experiência consistiu em acessar 5 *links* pré-definidos durante 5 minutos. Desses *links* 3 eram bloqueados pelos seus endereços de domínio. Aleatoriamente foram atribuídos aos usuários dispositivos de acesso de mesa (e.g., *desktop*) e celulares, de modo a demandar, em alguns casos, a conversão da linguagem de marcação. Foi aplicada uma carga progressiva de usuários até o limite de 20 usuários, com cada usuário clicando num link a cada 5 segundos.

Foram definidos dois cenários de teste: o primeiro, utilizando-se ontologias, com a descoberta e a composição de serviços realizados sob demanda, onde se obteve um tempo médio de resposta de 453 ms; o segundo, sem a utilização de ontologias, com a descoberta e composição de serviços definidos manualmente, com um tempo médio de 39 ms.

Do ponto de vista quantitativo, analisando os resultados da Figura 8, fica evidente o aumento no tempo de resposta e na demanda por processamento, que foram causados pelo uso de ontologias no FACI reconstruído. Entretanto, do ponto de vista qualitativo, fica evidente também a flexibilidade e precisão, que o uso de semântica no mecanismo de descoberta e composição proporciona ao engenheiro de software [Yao, Su e Yang 2006]. É necessário destacar também que esse tempo de resposta tende a diminuir com o amadurecimento das *API´s* semânticas. Além disso, esse estudo de caso demonstra que a opção, de realizar a pesquisa e a composição de serviços internamente,

apresentou um ganho significativo no tempo de resposta (de 40 a 50%) em relação a outros estudos de caso que usam *UDDI's* adaptadas [Naumenko et al. 2005][Srínívasan Paoluccí e Sycara 2005].

## 5. Trabalhos Correlatos

O uso de ontologias, para a descrição de contexto e de serviços de adaptação, é uma linha de pesquisa recente, devendo receber um grande impulso quando novas especificações de padrões, que suportam as tecnologias de Serviços *Web* (e.g., *UDDI*, *WSDL*), proverem suporte nativo para as descrições semânticas.

Dentre os trabalhos correlatos destaca-se o de [Berhe, Brunie e Pierson 2004], que trata da especificação formal do processo de adaptação e da utilização de *proxies* de adaptação de conteúdo e o do protocolo *ICAP*, mas que não emprega ontologias. Em [Zahreddine e Mahmoud 2005] é apresentada uma solução baseada em *OWL-S* e *WSDL*, para adaptação de conteúdo, com foco no desenvolvimento de um módulo que substitui o *UDDI*, visando ganhos de desempenho. Em [Srínívasan, Paoluccí e Sycara 2005] é proposta uma *API*, que proporciona um mapeamento entre as informações de registro da *UDDI* e as descrições semânticas do *OWL-S*. Apesar das vantagens proporcionadas por essa API que permite o uso, de semântica pelos servidores *UDDI* distribuídos pela Internet, o estudo de caso apresenta valores de tempo de resposta que inviabilizam, no atual estágio, o seu uso no domínio da adaptação de conteúdo. Em [Geyter e Soetens 2005] é proposto um modelo, baseado em redes de tarefa hierárquica, para a composição e a descoberta de serviços, e uma nova especificação para a descrição de regras muito similar às definidas pelo *SWRL* e *OWL-S*.

Apesar da qualidade desses trabalhos correlatos, a pesquisa apresentada neste artigo destaca-se pelo uso de ontologias para a descrição de perfis, proporcionando uma definição mais precisa do contexto, que resulta em uma melhoria na busca de serviços de adaptação. A utilização de serviços *Web* simplifica o desenvolvimento de servidores de adaptação e facilitará a migração da solução proposta para os futuros padrões de *Web* semântica. Ambos, ontologias e serviços *Web*, foram incorporados a uma abordagem baseada em um *framework* de componentes. A sua característica modular oferece uma solução flexível, previamente testada, para o domínio de adaptação de conteúdo na Internet, facilitando assim a reutilização e a manutenção das aplicações.

## 6. Conclusão

Este artigo propõe o emprego de ontologias e serviços *Web* em um ambiente ubíquo, concentrando-se na solução de problemas relacionados com a adaptação de conteúdo em dispositivos móveis, tais como celulares e *palmtops*.

Tendo em vista que é essencial a definição de uma política de adaptação e que esta é fundamentada num conjunto de informações, expressas através de perfis, e num conjunto de regras de adaptação, foi proposto que esses perfis e essas regras fossem descritos através de ontologias, utilizando-se uma linguagem apropriada para as suas especificações. Essa linguagem apresenta uma fácil integração com linguagens orientadas a objeto, auxiliando na implementação e interoperabilidade da proposta. Também foi adicionado um suporte a serviços *Web*, com a integração do *SOAP e WSDL*, de modo a utilizar a infra-estrutura de ferramentas e serviços disponíveis, facilitando assim o crescimento do número de servidores de adaptação.

Para viabilizar esta proposta foi desenvolvida uma extensão do FACI, a partir da adição, adaptação e reuso de componentes. Essa extensão resultou em um *framework* mais genérico, facilitando o desenvolvimento de novas soluções no domínio da adaptação de conteúdo.

Para trabalhos futuros sugere-se: o aprimoramento do módulo de *Matching and Composition* para melhorar a performance na descoberta e composição de serviços e também possibilitar que essa tarefa possa ser realizada externamente, utilizando-se as novas especificações de serviços *Web* Semânticos que estão sendo desenvolvidas (e.g., *OWL-S/UDDI*, *WSDL-S*); a adição de um componente que gerencie a troca de perfis de serviços entre *Adaptation Proxies*; e o desenvolvimento de novos *local* e *remote adapters* de modo a ampliar a variedade de adaptações disponíveis.

## 7. Referências

Baader, F. et al. (2003) "Volume Handbook on Ontologies in Information Systems of International Handbooks on Information Systems", pp. 3-31, Steffen Staab and Rudi Studer, Eds., Springer.

Berhe, G.; Brunie, L.; Pierson, J. (2004) "Modeling Service-Based Multimedia Content Adaptation in Pervasive Computing", Conference of Computing Frontiers, pp. 60-69.

Berners-Lee, T.; Hendler, J.; Lassila, O. (2001) "The Semantic Web", Scientific American, edição de maio de 2001.

Cannistrà, F. (2003) "Exploiting Ontologies to Achieve Semantic Convergence Between Different CC/PP-like RDF Schemes for Representing Device´s Capabilities: the SADiC Approach", Second International Semantic Web Conference (ISWC-03).

Claudino, R. A. T.; Souza, W. L.; Prado, A. F. (2005) "Um framework baseado em componentes para o domínio de adaptação de conteúdo na Internet", Anais do 19º Simpósio Brasileiro de Engenharia de Software, pp. 88-103, Uberlândia, MG.

Elson, J.; Cerpa, A. (2003) "Internet Content Adaptation Protocol", IETF Request for Comments 3507, <http://www.isi.edu/in-notes/rfc3507.txt>.

EODM. (2006) "IBM Integrated Ontology Development Toolkit". <http://www.alphaworks.ibm.com/tech/semanticstk>.

Geyter, M.; Soetens, P. (2005) "A Planning Approach to Media Adaptation within the Semantic Web", DMS 2005, Banff, Canada.

Guarino, N. (1998) "Formal Ontology and Information Systems" in: N. Guarino, (Ed.) Formal Ontology in Information Systems. pp. 3-15, IOS Press, Netherlands.

Hansmann, U. et al. (2003) "Pervasive Computing", Springer-Verlag, Second Edition.

Horrocks, I. et al. (2003) "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", DAML, <http://www.daml.org/2003/11/swrl/>.

Jena. <http://jena.sourceforge.net/>.

Klyne, G. et al. (2004) "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", W3C, <http://www.w3.org/TR/CCPP-struct-vocab/>.

Knublauch, H. et al. (2006) "A Semantic Web Primer for Object-Oriented Software Developers", <http://www.w3.org/TR/2006/NOTE-sw-oosd-primer-20060309/>.

Lara, R. et al. (2005) "Deliverable 2.4. Semantics for Web Service Discovery and Composition", KnowledgeWeb.

Martin, D. (2004) "OWL-S: Semantic Markup for Web Services", DAML, <http://www.daml.org/services/owl-s/1.1/>.

Martin, D. L.; Mcllraith, S. A. (2003) "Bringing Semantics to Web Services", IEEE Intelligent Systems, IEEE Press, pp. 90-93, USA.

McGuinness, D. L.; Harmelen, F. V. (2004) "OWL Web Ontology Language Overview", W3C, < http://www.w3.org/TR/owl-features/>.

Monola, F.; Miller, E. (1999) "Resource Description Framework (RDF) Model and Syntax Specification", W3C , <http://www.w3.org/TR/REC-rdf-syntax/>.

Naumenko A. et al. (2005) "Using UDDI for Publishing Metadata of the Semantic Web", In: M. Bramer and V. Terziyan (Eds.): Industrial Applications of Semantic Web, Proceedings of the 1st International IFIP/WG12.5 Working Conference IASW-2005, Jyväskylä, Finland, Springer, IFIP, pp. 141-159.

OWL-S. <http://www.mindswap.org/2004/owl-s/api/>.

Pellet. <http://www.mindswap.org/2003/pellet/>.

Prud'hommeaux, E.; Seaborne, A. (2006) "SPARQL Query Language for RDF", W3C, <http://www.w3.org/TR/2006/WD-rdf-sparql-query-20060220/>.

Semantic Web Activity. <http://www.w3.org/2001/sw/>.

SOAP. "Simple Object Access Protocol", W3C, <http://www.w3.org/TR/soap/>.

Srínívasan N., Paoluccí M., Sycara K., (2005) "Adding OWL-S to UDDI, implementation and throughput". Robotics Institute, Carnegie Mellon University, USA.

WSDL. "Web Service Definition Language", W3C, <http://www.w3.org/TR/wsdl>

Yao, Y.; Su, S.; Yang, F. (2006) "Service Matching Based on Semantic Descriptions", In: Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW), Guadeloupe, French Caribbean, pg. 126-131.

Zahreddine, W.; Mahmoud Q. H. (2005) "A Framework for Automatic and Dynamic Composition of Personalized Web Services", AINA 2005, pp. 513-518.