EDGARDO PAUL PONCE ESCOBEDO

MODELAGEM DE CONTEXTO UTILIZANDO ONTOLOGIAS

São Paulo

2008

EDGARDO PAUL PONCE ESCOBEDO

MODELAGEM DE CONTEXTO UTILIZANDO ONTOLOGIAS

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do Título de Mestre em
Engenharia Elétrica.

São Paulo

2008

EDGARDO PAUL PONCE ESCOBEDO

MODELAGEM DE CONTEXTO UTILIZANDO ONTOLOGIAS

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do Título de Mestre em
Engenharia Elétrica.

Área de Concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Sergio Takeo Kofuji

São Paulo

2008

AGRADECIMENTOS

Agradeço a Deus e aos meus pais Vicente E. e Elva E. pelo carinho e apoio constante em cada momento da minha vida.

A meu orientador, Prof. Dr. Sergio Takeo Kofuji, pela orientação e estímulo no desenvolvimento deste trabalho.

Aos meus amigos do GPP, PUR e a todos que me brindaram sua amizade e apoio.

Aos meus colegas do grupo PAD, pelo seu companheirismo.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

Ao pessoal do Intermedia Lab da USP-São Carlos, pela sua colaboração, e a todos aqueles que colaboraram, de forma direta e indireta, na execução desta dissertação.

RESUMO

Com os avanços dos processos da microeletrônica temos dispositivos menores e com maior poder de computação e comunicação. Um Ambiente Pervasivo contém diferentes dispositivos, tais como sensores, atuadores, eletroeletrônicos e dispositivos móveis que interagem com a pessoa de forma natural ao conhecer o contexto. A diversidade de dispositivos e informações do Ambiente Pervasivo introduz um problema de interoperabilidade. Um Ambiente Pervasivo é dinâmico devido à mobilidade do usuário, a variedade de dispositivos. Neste trabalho, é proposto um modelo semântico de contexto para permitir interoperabilidade e fornecer suporte ao dinamismo do Ambiente Pervasivo. O modelo proposto contém características da modelagem de contexto realizadas por trabalhos anteriores, assim como sua integração com a modelagem de preferências das pessoas, políticas de privacidade e serviços. Verificou-se que o modelo de contexto proposto é adequado mediante sua aplicação em um Estudo de Caso e mediante testes realizados. Mostra-se que a modelo de contexto utilizado ontologias e Serviços Web Semânticos permite tratar com informação incompleta e inconsistente, bem como fornece suporte na interoperabilidade e ao dinamismo do Ambiente Pervasivo.

Palavras-chave: Computação Pervasiva, Computação Ciente de Contexto, Web Semântica, Serviços Web Semânticos.

ABSTRACT

Advances in microelectronic processes have allowed smaller devices with more computation and communication power. Pervasive environment contains different devices like electronic sensor, actuators and mobile devices which interact with the person naturally after the context is known. The device and information diversity introduce an interoperability problem. Pervasive environments are dynamics because of user's mobility and a variety of devices. In this work, we propose a context model to allow interoperability and to give support to pervasive environment dynamism. The proposed model contains features of context modeling developed in previous works, as well as, their integration with the modeling of the people's preferences, privacy policies and services. It was verified that the context model is appropriate by their application in a Case Study and by accomplished tests. It is shown that the model of context using ontologies and Semantic Web Services allow us to work with inconsistent and incomplete information, as well as gives support to interoperability and dynamism of the Pervasive Environment.

Keywords: Pervasive Computing, Context-Aware Computing, Semantic Web, Semantic Web Services.

LISTA DE FIGURAS

Figura 1 - Características de um Ambiente Pervasivo	9
Figura 2 - MUSDAC em um shopping, um ambiente B3G. Extraído de (RAVERDY; ARMAND; ISSARNY, 2006).....	11
Figura 3 - Diferentes técnicas para interação com objetos físicos. Extraído de (BROLL et al., 2007)	14
Figura 4 - OWL-S Ontologia de serviços. Extraído de (MARTIN et al., 2004)	25
Figura 5 - Visão geral das ontologias do OMC	28
Figura 6 - Visualização gráfica da ontologia Person	31
Figura 7 - Visualização gráfica da ontologia Space	33
Figura 8 - Visualização gráfica da ontologia geoRelations	35
Figura 9 - Visualização gráfica da ontologia Device	36
Figura 10 - Visualização gráfica da ontologia ComputationalDevice	37
Figura 11 - Visualização gráfica da ontologia HomeDevice	38
Figura 12 - Visualização gráfica da ontologia Activity	40
Figura 13 - Diagrama de atividades do serviço “Ingressar na casa”	46
Figura 14 - Fluxo de Trabalho do processo “Autorizar Pessoa”	51
Figura 15 - Fluxo de Trabalho do processo “Verificar parâmetros contextuais”	52
Figura 16 - Mapa de um espaço geográfico da Casa Inteligente.....	57
Figura 17 - Janela dos dados inferidos do Teste - 1 utilizando o Pellet 4.0	70
Figura 18 - Diagrama de atividades do serviço “Ingressar na casa”	73

Figura 19 - Listado das classes da ontologia HomePerson	95
Figura 20 - Listado das propriedades tipo de objeto da ontologia HomePerson	96
Figura 21 - Listado das classes da ontologia HomePerson	97
Figura 22 - Listado das instancias da ontologia HomePerson	98
Figura 23 - Livrarias necessárias para a implementação do Estudo de Caso	99
Figura 24 - Pasta das ontologias do OMC para a implementação do Estudo de Caso	100

LISTA DE QUADROS

Quadro 1 - Descrição de um serviço em OWL-S.....	48
Quadro 2 - Descrição do dispositivo “Porta”	48
Quadro 3 - Descrição dos IOPEs do perfil do serviço “Abrir Porta”	49
Quadro 4 - Definição dos IOPEs do processo do serviço “Abrir Porta”.....	50
Quadro 5 - Parâmetros contextuais do processo composto “Autorizar Pessoa”	51
Quadro 6 - O processo composto “Obter parâmetros contextuais”	52
Quadro 7 - Parte de código da ontologia Person	54
Quadro 8 - Propriedades de relacionamentos da ontologia HomePerson	55
Quadro 9 - Modelagem em OWL dos relacionamentos de Ricardo.....	55
Quadro 10 - Definição da classe “pessoa adulta”	56
Quadro 11 - Descrição dos objetos contidos na cozinha	58
Quadro 12 - Relações espaciais entre a pessoa e a mesa	59
Quadro 13 - Definição de entidades temporais	60
Quadro 14 - Restrição do tempo de entrada á casa a um intervalo de tempo permitido.....	60
Quadro 15 - Descrição da atividade “Rodrigo Assiste Televisão”	61
Quadro 16 - Política de privacidade para ingressar à casa.....	62
Quadro 17 - Regra SWRL para controlar o ingresso na casa somente a residentes	63
Quadro 18 - Preferência de Rodrigo por assistir televisão	64

Quadro 19 - Preferência de Rodrigo por assistir televisão nas tardes e em seus tempos livres	64
Quadro 20 - Definição de equivalência entre duas propriedades de ontologias distintas	73
Quadro 21 - Instâncias exemplo da localização de uma pessoa e um objeto	74
Quadro 22 - Instâncias exemplo dos dados de uma pessoa e sua localização.....	74
Quadro 23 - Parte de código em Java para carregar uma ontologia	101
Quadro 24 - Parte de código em Java para realizar inferência	101

LISTA DE TABELAS

Tabela 1 - Categorização de contexto centrado na pessoa e no ambiente	18
Tabela 2 - Parâmetros contextuais do Serviço “Abrir Porta”	49
Tabela 3 - Descrição de dispositivos para o Estudo de Caso	62
Tabela 4 - Instâncias exemplo da modelagem de pessoas.....	69
Tabela 5 - Propriedades tipo objeto e classes de Ricardo	69
Tabela 6 - Diferentes abordagens para modelagem de contexto	79
Tabela 7 - Termos para a modelagem de pessoas de uma Casa Inteligente.....	94

LISTA DE ABREVIATURAS E SIGLAS

B3G	Beyond 3G
CoBrA	Context Broker Architecture
CML	Context Modelling Language
CPU	Central Processing Unit
DL	Description Logics
Foaf	Friend of a Friend
GTSH	Gator Tech Smart House
GUI	Graphical User Interface
HCI	Human Computer Interaction
OMC	Ontologias para Modelagem de Contexto
OSGi	Open Service Gateway Initiative
OWL	Ontology Web Language
PC	Personal Computer
PDA	Personal Digital Assistant
RDF	Resource Description Language
SeCoM	Semantic Context Model
SO	Sistema Operacional
SOUPA	Standard Ontology for Ubiquitous and Pervasive Applications
SWRL	Semantic Web Rule Language

SWS	Serviços Web Semânticos
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
XML	EXtensible Markup Language

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1. Motivação	1
1.2. Objetivos	4
1.3. Metodologia.....	5
1.4. Organização do trabalho.....	6
2. CONCEITOS	8
2.1. Computação Pervasiva	8
2.2. Computação ciente de contexto	15
2.3. Web Semântica.....	20
3. MODELO SEMÂNTICO DE CONTEXTO	27
3.1. Modelagem de contexto.....	27
3.2. Visão geral	28
3.3. Pessoas.....	30
3.4. Espaço	32
3.5. Dispositivos	36
3.6. Tempo	39
3.7. Atividades.....	40
3.8. Políticas de privacidade	41
3.9. Preferências	42

3.10. Serviços cientes de contexto	43
4. ESTUDO DE CASO.....	44
4.1. Enunciado	44
4.2. Descrição dos serviços	45
4.3. Modelagem de serviços cientes de contexto.....	47
4.4. Modelagem do contexto.....	53
5. RESULTADOS	65
5.1. Materiais.....	65
5.2. Testes.....	67
5.3. Comparação com outras abordagens.....	75
6. CONCLUSÕES E TRABALHOS FUTUROS.....	82
6.1. Conclusões.....	82
6.2. Limitações	82
6.3. Trabalhos futuros	83
6.4. Publicações	83
REFERÊNCIAS.....	85
APÊNDICE A - ROTEIRO PARA A ELABORAÇÃO DE UM ESTUDO DE CASO	93

1. INTRODUÇÃO

1.1. Motivação

Com o avanço da microeletrônica temos dispositivos menores e com maior poder de computação e comunicação. A visão da Computação Pervasiva ou Ubíqua é endereçada ao aumento do número de dispositivos e tecnologias em nosso ambiente. Segundo esta visão (WEISER, 1995) os recursos de computação serão onipresentes na vida diária e estarão interconectados com a finalidade de fornecer informação e/ou serviços aos usuários em qualquer lugar e momento.

A Computação Pervasiva proporciona uma interação natural entre a pessoa e o ambiente, a qual requer uma mínima intervenção humana e acontece de forma autônoma, interativa e relevante (SATYANARAYANAN, 2001). Neste cenário, nós realizamos nossas atividades livremente, sem nos preocupamos em dar suporte à tecnologia (EDWARDS, 2006).

Ambientes Pervasivos demandam aplicações que sejam capazes de operar em condições altamente dinâmicas e demandam pouca atenção das pessoas. Portanto, para que as aplicações de Computação Pervasiva atendam a essas necessidades elas têm que ser cientes de contexto ou sensíveis ao contexto (HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002).

O contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade que é considerada relevante entre o usuário e a aplicação (DEY; SALBER; ABOWD, 2001). Aplicações cientes de contexto requerem uma infraestrutura para aquisição, gerenciamento e disseminação da informação de contexto (HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002).

Strang e Linnhoff-Popien (2004) afirmam que um modelo de contexto apropriado é a chave principal para qualquer sistema ciente de contexto. Os autores realizaram

uma análise na qual se destaca o uso de ontologias como a técnica mais promissora para a modelagem de contexto.

Um Ambiente Pervasivo tem uma natureza dinâmica, devido à mobilidade do usuário, a variedade de dispositivos e tecnologias existentes, assim como às mudanças constantes nos perfis dos usuários (ZHOU et al., 2007). Para fornecer suporte ao dinamismo do Ambiente Pervasivo, requer a definição das suas regras de comportamento em tempo de execução (WALTENEGUS, 2006).

A modelagem de contexto utilizando ontologias permite a definição do comportamento do Ambiente Pervasivo em tempo de execução, mas não fornece o suporte necessário para lidar com o dinamismo do ambiente. Uma das alternativas para lidar com o dinamismo é o uso de Serviços Web Semânticos (SWS).

Considerando que os SWS possibilitam uma ampla faixa de novas tarefas de automatização, incluindo composição automática, invocação e descoberta de serviços. Em um trabalho anterior (PONCE et al, 2007) foi apresentada uma abordagem que combina a modelagem de contexto e a modelagem de serviços utilizando SWS. Esta abordagem permite que o Ambiente Pervasivo, possa se adaptar as mudanças do ambiente, como a disponibilidade de dispositivos, permitindo a descoberta, invocação, e composição de serviços automaticamente.

Broens et al. (2004) e Mokhtar et al. (2006) descrevem um algoritmo de descoberta semântica de serviços utilizado ontologias. Os autores modelaram serviços semanticamente e as complementaram com informações de contexto. Porém esses autores não consideraram a utilização de um modelo de contexto para Ambientes Pervasivos.

A contribuição principal deste trabalho é o desenvolvimento de um novo modelo de contexto para Ambientes Pervasivos, o OMC. Existem trabalhos que modelam contexto utilizando ontologias (CHEN; FININ; JOSHI, 2004) (GU et al., 2004) (BULCÃO; PIMENTEL, 2005). Foram estudadas essas abordagens e verificou-se que, essas abordagens não permitem à modelagem de alguns aspectos presentes em casos reais, como as preferências políticas de privacidade e suporte ao dinamismo do Ambiente Pervasivo.

O OMC apresenta as seguintes características:

- permite o gerenciamento de informações altamente relacionadas e incompletas, devido à utilização de semântica que já foi apresentado por outros trabalhos (CHEN; FININ; JOSHI, 2004) (GU et al., 2004) (BULCÃO; PIMENTEL, 2005);
- apresenta ontologias mais robustas para a modelagem de espaço e dispositivos. Para a modelagem de espaços foram construídas as ontologias **Space**, **HomeSpace**, **geoDescription**, **geoCoordinateSystem** e **GeoRelations** as quais reutilizam conceitos de **GeoOntology** (LING et al., 2006) (GEOONTOLOGIES ..., 2004) e wgs84_pos¹ (BRICKLEY, 2003). Para a modelagem de dispositivos foram construídas as ontologias **Device**, **ComputationalDevice** e **HomeDevice**. As duas últimas reutilizam os conceitos e definições das ontologias do Projeto *Amigo*²;
- inclui a definição e inferência de políticas de privacidade em *Semantic Web Rule Language Ontology Web Language* (SWRL). Henricksen e Indulska (2006) apresentam a modelagem de preferências, seu modelo de contexto não utiliza ontologia, o que dificulta a interoperabilidade e a inferência semântica;
- inclui a definição e inferência de preferências em SWRL. Chen, Finin e Joshi (2005) apresentam o Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA). O SOUPA utiliza a linguagem de políticas de privacidade Rei (REI ..., 2005) para definir: direitos, proibições e obrigações. Essa linguagem é robusta, mas não é definida em *Ontology Web Language* (OWL) ou em SWRL, o que dificulta a inferência semântica aplicada com atuais máquinas de inferência como o Pellet (PELLET ..., 2008);
- fornece suporte ao dinamismo do Ambiente Pervasivo. A modelagem dos serviços do Ambiente Pervasivo utilizando SWS. O OMC integra a modelagem de contexto utilizando OWL e a modelagem de serviços utilizando

1 O Wgs84_pos é um vocabulário básico em RDF para representar latitude, longitude e outra informação para localizar espacialmente coisas, utilizando o sistema de coordenadas WGS84.

2 O projeto Amigo (IST Amigo Project) (AMIGO ..., 2008) é um projeto europeu, que tem como objetivo alcançar o potencial pleno das Casas Inteligentes (Home Networking) para melhorar a vida das pessoas.

a ontologia de serviços OWL-S (PONCE et al, 2007) e segundo o nosso conhecimento essa integração não foi apresentada anteriormente.

1.2. Objetivos

1.2.1. Objetivo

Os objetivos iniciais deste trabalho foram os seguintes:

- criar um modelo de contexto adequado para Ambientes Pervasivos;
- construção de uma interface gráfica visual simples do Aml para controle, monitoração, programação e prototipagem do sistema;
- servir como base de integração com dispositivos reais como: rede de sensores sem fio, interação com um dispositivo móvel, entre outros.

Devido à complexidade dos objetivos propostos anteriormente, o objetivo deste trabalho é o desenvolvimento, a aplicação e a análise de um modelo de contexto adequado para Ambientes Pervasivos.

1.2.2. Objetivos específicos

Os objetivos específicos são os seguintes:

- modelar o contexto em Ambientes Pervasivos utilizando ontologias. Na modelagem de contexto devem ser consideradas políticas de privacidade e a modelagem de preferências;

- modelar serviços em Ambientes Pervasivos utilizando SWS e integrar com a modelagem de serviços;
- desenvolver um Estudo de Caso no qual seja utilizado o modelo de contexto proposto;
- avaliar e analisar os benefícios da modelagem de contexto utilizando ontologias;
- comparar o modelo de contexto proposto com outras abordagens.

1.3. Metodologia

A seguir é descrita a metodologia utilizada neste trabalho.

1.3.1. Modelo de contexto

Foi desenvolvido um modelo semântico de contexto para Ambientes Pervasivos denominado Ontologias para Modelagem de Contexto (OMC) (Capítulo 3) o qual considera os seguintes elementos:

- este modelo tem uma divisão em dois níveis. No primeiro nível apresentam-se as ontologias independentes de domínio e no segundo as dependentes de domínio;
- a linguagem OWL (SMITH; WELTY; MCGUINNESS, 2004) no dialeto DL (*Description Logics*) é utilizada na modelagem de contexto.
- as regras na linguagem SWRL (HORROCKS et al., 2004) são utilizadas para descrever as preferências das pessoas e as políticas de privacidade;
- a ontologia de serviços OWL-S (MARTIN et al., 2004) é utilizada na modelagem de serviços no Ambiente Pervasivo. A modelagem dos serviços

semânticamente possibilita a automatização da descoberta e composição destes.

1.3.2. Estudo de Caso

Foi aplicado o modelo de contexto OMC em Estudo de Caso cujo domínio é uma Casa Inteligente (Capítulo 4). De acordo com o Estudo de Caso proposto, são modelados os serviços e o contexto semanticamente.

1.3.3. Resultados

Na obtenção dos resultados foram realizados testes (Seção 5.2 e Seção 5.3) para avaliar a utilização da inferência semântica no OMC, assim como foi comparado o modelo OMC com outras abordagens (Seção 5.4).

1.4. Organização do trabalho

A estrutura dos capítulos da dissertação é a seguinte:

- capítulo 2: Conceitos. Esse capítulo apresentará os conceitos no estado da arte sobre Computação Pervasiva, Computação Ciente de Contexto, Web Semântica e SWS;
- capítulo 3: Modelo semântico de contexto. Nesse capítulo será descrito o modelo de contexto OMC. Serão descritas também as ontologias construídas para modelagem de contexto e a integração com a modelagem semântica de serviços;

- capítulo 4: Estudo de Caso. Nesse capítulo será descrito um Estudo de Caso no domínio de uma Casa Inteligente. No desenvolvimento do Estudo de Caso será utilizado o OMC;
- capítulo 5: Resultados. Esse capítulo apresenta os materiais utilizados e os testes realizados para avaliar a utilização de Web Semântica na modelagem de contexto. Assim como também contém a comparação do modelo de contexto proposto com outros trabalhos;
- capítulo 6: Conclusões e trabalhos futuros;
- Apêndice A: Roteiro para a elaboração de um Estudo de Caso.

2. CONCEITOS

Este capítulo apresenta os conceitos no estado da arte sobre Ambientes Pervasivos e suas características. É destacada principalmente a característica ciente de contexto por ser o foco principal deste trabalho. Este capítulo também apresenta conceitos sobre Web Semântica.

2.1. Computação Pervasiva

2.1.1. Definições

A visão da Computação Pervasiva ou Ubíqua endereça o aumento do número de dispositivos e tecnologias em nosso ambiente natural. Segundo esta visão (WEISER, 1995) os recursos de computação serão onipresentes na vida diária e estarão interconectados com a finalidade de fornecer informação e/ou serviços aos usuários em qualquer lugar e momento.

A Computação Pervasiva proporciona uma interação natural entre a pessoa e o ambiente, a qual requer uma mínima intervenção humana e acontece de forma autônoma, interativa e relevante (SATYANARAYANAN, 2001). Este tipo de interação entre as pessoas e seu ambiente é um dos maiores desafios da Computação Pervasiva, cujo interesse é de adequar-se a tecnologia a nossas vidas diárias (WEISER, 1995).

A Computação Pervasiva tem o potencial de mudar a forma como desenvolvemos as nossas atividades cotidianas (HARIHAR; KURKOVSKY, 2005). Desse modo em um Ambiente Pervasivo não se percebe que as pessoas estão interagindo com as máquinas.

2.1.2. Características

Segundo Abowd e Mynatt (2000), a Computação Pervasiva promete mais do que uma infra-estrutura, mas acima de tudo, novos paradigmas de interação inspirados pelo amplo acesso à informação e às capacidades computacionais. Os autores assinalam três desafios ou características nos Ambientes Pervasivos, sendo eles: i) interfaces naturais; ii) ciente de contexto; e iii) captura e acesso automatizado.

Outras divisões de características dos Ambientes Pervasivos são dadas por Harihar e Kurkovsky (2005):

- acesso Ubíquo;
- comportamento inteligente;
- interação natural;
- ciente de contexto.

Os autores apontam também que além dessas características, os Ambientes Pervasivos devem ser confiáveis e seguros.

Na Figura 1, são apresentadas as quatro características de um Ambiente Pervasivo propostas por Harihar e Kurkovsky (2005).

Figura 1 - Características de um Ambiente Pervasivo

A seguir, serão detalhadas essas características, sendo que a característica ciente de contexto será apresentada com maior detalhe na Seção 2.2 por ser o foco deste trabalho.

2.1.2.1. Acesso ubíquo

Acesso ubíquo refere-se ao acesso à informação e aos serviços do ambiente em qualquer lugar e momento. Segundo Harihar e Kurkovsky (2005), uma diferença importante entre a Computação Tradicional e a Computação Pervasiva está no acesso à informação e aos serviços. No paradigma tradicional, o usuário realiza manualmente algumas tarefas, mas na Computação Pervasiva, o usuário acessa de forma mais natural ou transparente (GRIMM et al., 2004).

Devido ao dinamismo do Ambiente Pervasivo, é necessária uma tecnologia robusta que possa integrar com os dispositivos de uma maneira uniforme (HARIHAR; KURKOVSKY, 2005). Como exemplo, temos os protocolos de localização dinâmica de serviços¹ (EDWARDS, 2006).

A seguir, são descritas algumas arquiteturas *middleware*² utilizadas para fornecer acesso ubíquo como: Jini (JINI ..., 2008), *Open Service Gateway Initiative* (OSGi) (OSGI ..., 2008) e MUSDAC (RAVERDY; ARMAND; ISSARNY, 2006).

O Jini (JINI ..., 2008), é um protocolo de localização dinâmica de serviços que pode ser utilizado para a construção de sistemas de redes adaptativas, sendo elas: escaláveis, adaptáveis e flexíveis. Essas características são requeridas em ambientes de computação dinâmicos.

1 Um protocolo de localização dinâmica de serviços (service discovery protocol), permite localizar dinamicamente em uma rede um serviço ou dispositivo desejados.

2 O Middleware é desenvolvido acima do sistema operacional e as redes fornecendo aos desenvolvedores de aplicação um alto nível de abstração, a qual oculta a complexidade introduzida pela distribuição (CAPRA; EMMERICH; MASCOLO, 2001).

Harihar e Kurkovsky (2005) discutem o uso da tecnologia de redes Jini para possibilitar e integrar tecnologias usadas por Ambientes Pervasivos. Já Edwards (2006) realiza uma comparação do Jini e outros protocolos de localização dinâmica de serviços em Ambientes Pervasivos.

Outro *middleware* para Ambientes Pervasivos é o OSGi, o qual foi construído em 1999 e define uma especificação livre para o envio e o fornecimento de múltiplos serviços em residências, automóveis e em outros ambientes.

O Gator Tech Smart House (GTSH) (HELAL et al., 2005) é um projeto da Universidade da Flórida cujo objetivo é de criar Ambientes Pervasivos de última geração apresentando uma arquitetura *middleware* aplicável para qualquer Ambiente Pervasivo. A arquitetura proposta do GTSH para desenvolver Ambientes Pervasivos utiliza o middleware OSGi e a modelagem de contexto com ontologias.

O MUSDAC é uma plataforma *middleware* que interopera com protocolos existentes de descoberta dinâmica de serviços. O MUSDAC permite aos clientes descobrir e interagir com serviços implementados em ambientes B3G¹ (*Beyond 3G*) (RAVERDY; ARMAND; ISSARNY, 2006). Este trabalho considera um ambiente B3G como o Ambiente Pervasivo definido por Weiser (1995). A Figura 2 mostra como em um shopping, mediante o MUSDAC, um cliente pode se conectar a varias redes em um ambiente B3G.

Figura 2 - MUSDAC em um shopping, um ambiente B3G. Extraído de (RAVERDY; ARMAND; ISSARNY, 2006)

1 B3G (Beyond 3G) Além da Terceira geração de tecnologia sem fio.

2.1.2.2. *Comportamento inteligente*

A característica comportamento “Inteligente” de um Ambiente Pervasivo, refere-se à sua habilidade de se adaptar ao comportamento do usuário de uma forma personalizada para suprir aos usuários informação no tempo e lugar correto (HARIHAR; KURKOVSKY, 2005).

Em seguida são citados trabalhos que utilizam aprendizagem de máquina, agentes de software e SWS, como técnicas que possibilitam um comportamento “inteligente” em Ambientes Pervasivos.

Krause et al. (2006), utilizam técnicas de aprendizagem de máquina e análise estatística em um dispositivo móvel ciente de contexto. Os autores desenvolveram um protótipo SenSay que utiliza vários sensores vestíveis (*wearables*) para obter a atividade e estado psicológico da pessoa.

Kallio et al. (2006) descrevem um método visual para a animação dos movimentos das mãos gerados por acelerômetros. Os acelerômetros produzem séries de tempo executadas pelos movimentos das mãos. Os autores utilizam *Hidden Markov Models* (HMM) para a modelagem de tais séries de tempo. HMM são máquinas de estados finitos capazes de gerar e analisar uma série de tempo mediante um modelo probabilístico.

Zhou et al. (2007) apresentam uma arquitetura baseada em agentes com o nome MDAgent. O MDAgent utiliza duas funcionalidades dos agentes de software: autonomia e mobilidade. O agente autônomo é responsável pelo raciocínio e a tomada de decisão de acordo com a informação de contexto, enquanto que o agente móvel é responsável pela migração de componentes de aplicação de acordo com a nova localização da pessoa. O MDAgent utiliza ontologias na modelagem de contexto.

A aplicação dos Serviços Web Semânticos possibilita uma nova faixa de tarefas de automação, as quais são executadas por pessoas, como a descoberta automatizada, invocação e composição de serviços (MCILRAITH; MARTIN, 2003).

Broens et al. (2004) utilizam SWS e informação de contexto. Esse trabalho foca-se na descrição do algoritmo *matching* que utiliza atributos contextuais. Mokhtar et al. (2005) apresentam uma proposta para a composição de serviços cientes de contexto em Ambientes Pervasivos. Os autores modelam serviços e tarefas do usuário em OWL-S e as complementam com informação de contexto.

2.1.2.3. *Interação natural*

A interação natural em Ambientes Pervasivos refere-se a uma interação homem-computador (*Human Computer Interaction*) (HCI) de forma quase ou se possível natural. Interagimos com o mundo real com as mãos, os gestos, as palavras e com o mundo digital usando interfaces com o uso do mouse, teclado, entre outros.

As interfaces de computador que suportam mais de uma forma de comunicação natural humana (voz, gestos, *handwriting*) estão substituindo elementos do paradigma da interação *Graphical User Interface* (GUI) que majoritariamente utiliza o teclado e o mouse.

Um desafio na comunidade científica de Computação Pervasiva é a necessidade de gerenciar complexas interações entre numerosos dispositivos interconectados (LEE; HELAL; LEE, 2006). Para fornecer uma interação natural em um Ambiente Pervasivo é requerida a investigação de novas metáforas de interação HCI (KALLIO et al., 2006). A seguir apresentam-se trabalhos que utilizam diferentes metáforas de interação HCI para fornecer uma interação natural.

Broll et al. (2007) apresentam diferentes formas de interação de um dispositivo móvel com objetos físicos (*physical mobile interactions*). Os autores aumentaram as funcionalidades dos dispositivos móveis para a interação dos objetos do cotidiano. A Figura 3 apresenta diferentes formas de interação de um dispositivo móvel com objetos físicos:

- pelo toque de objetos físicos (*touching*) (Figura 3a), os objetos físicos são aumentados com tags *RFIDs*, os quais armazenam informação. Os usuários selecionam esses objetos pelo toque e lêem a informação que contem;
- pela sinalização dos objetos físicos (*pointing*) (Figura 3b), os objetos físicos são aumentados com marcadores visuais que podem ser capturados com câmeras de telefones moveis e interpretados por algoritmos de reconhecimento de imagens;
- localização (Figura 3c), uso de dispositivos GPS externos via Bluetooth para medir a proximidade de objetos físicos e interagir com eles apropriadamente;
- escanear (Figura 3d), usa as funcionalidades Bluetooth dos celulares de procurar, conectar-se e interagir com outros dispositivos e objetos aumentados na sua vizinhança.

Figura 3 - Diferentes técnicas para interação com objetos físicos. Extraído de (BROLL et al., 2007)

Alsos e Svanæs (2006) apresentam um estudo de comparação de técnicas de interação utilizando dispositivos *handhelds* em um hospital. Foi desenvolvidos e testados um conjunto de implementações protótipo em um cenário de pré-cirurgia.

Uma das formas naturais de interação é mediante gestos. Kallio et al. (2006) descrevem um método visual para a animação dos movimentos das mãos gerados por acelerômetros. A visualização dos gestos auxilia o usuário no controle dos gestos. O controle de gestos é importante devido às pessoas tenderem a executar gestos similares de diferentes formas.

2.2. Computação ciente de contexto

2.2.1. Definições

O contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade que é considerada relevante entre o usuário e a aplicação (DEY; SALBER; ABOWD, 2001). O contexto compreende informações sobre as pessoas (gostos, preferências, localização), os dispositivos (estado, disponibilidade) e o ambiente (temperatura, tempo, condições de meteorologia).

“Um sistema é ciente de contexto quando o sistema utiliza o contexto para fornecer informação relevante e/ou serviços ao usuário, onde a relevância depende das tarefas do usuário” (DEY; SALBER; ABOWD, 2001).

Um Ambiente Pervasivo cujo interesse é ser menos intrusivo ao usuário tem que ser ciente ao contexto (SATYANARAYANAN, 2001). Portanto a característica ciente de contexto é muito importante, pois com esta característica, o Ambiente Pervasivo tem a capacidade de interagir com a pessoa de forma adequada, gerenciando apropriadamente as informações e os recursos disponíveis.

2.2.2. Dimensões do contexto

Nos primeiros sistemas cientes de contexto foi considerada principalmente a informação de localização, mas existem outras informações de contexto. Schmidt, Beigl e Gellersen (1999) sugerem uma classificação das características do contexto. Essa classificação inclui os fatores humanos: o usuário (seu ambiente social e suas tarefas) e o ambiente físico (a localização, a infra-estrutura e as condições físicas).

Outra taxonomia sobre o contexto é proposta por Abowd e Mynatt (2000) a qual contém cinco divisões ou dimensões contextuais, sendo conhecidas como "*five W's*", pelas letras iniciais em inglês: *Who* (Quem), *What* (Que), *Where* (Onde), *When* (Quando), *Why* (Porque).

A seguir, serão descritas as dimensões contextuais apresentadas por Abowd e Mynatt (2000):

- Quem (*Who*): Informação referente a aquele ou aquilo que realiza uma ação. Pode ser uma pessoa, um dispositivo, um agente software - como um serviço, uma aplicação ou um sistema.
 - A pessoa é o elemento central de um Ambiente Pervasivo e a resposta do ambiente depende da pessoa. O sistema tem que identificar a pessoa e conhecer suas preferências e intenções;
 - Os dispositivos pervasivos são muito variados e têm funcionalidades diferentes;
 - Agentes computacionais, os quais incluem os serviços e as aplicações que realizam uma determinada ação.
- Onde (*Where*): Informação sobre a localização das pessoas e os objetos em um determinado espaço ou ambiente. Nesta dimensão estão incluídas também informações sobre as relações espaciais entre lugares, objetos e pessoas. Relações espaciais como, se duas pessoas se encontram próximas, se estão em um mesmo ambiente;
- Quando (*When*): Informações temporais das atividades das pessoas e os eventos que acontecem em um Ambiente Pervasivo. Informações como a

duração das atividades, o tempo que um determinado evento inicia e termina, definição das rotinas de uma pessoa através de um histórico das atividades dessa pessoa no qual é possível determinar suas preferências;

- O Que (*What*): Informações relacionadas às atividades, às ações feitas pelas pessoas ou dispositivos. Esta dimensão de contexto é mais complexa que as anteriores, pois esta relacionada com elas e não é facilmente fornecida por um sensor, sendo inferida ou derivada usando várias informações de contexto. Uma atividade é executada em um determinado espaço (onde) e momento (quando). De acordo com o domínio de aplicação existe um conjunto de atividades a serem efetuadas pelas pessoas (quem), por exemplo: cozinhar, caminhar, assistir televisão e estudar;
- Por que (*Why*): Esta dimensão é relacionada com o motivo pelo qual é feita uma determinada ação. As motivações das pessoas podem ser inferidas em geral combinando outras dimensões contextuais e considerando parâmetros da pessoa como seu estado emocional, seu batimento cardíaco, entonação vocal, entre outras. (BULCÃO; PIMENTEL, 2005).

2.2.3. Classificação da informação de contexto

Aquisição da informação de contexto: segundo Schilit, Adams e Want (1994) nas aplicações, o contexto é adquirido diretamente pela pessoa de forma explícita, ou implícita com a monitoração da pessoa e com as atividades baseadas no computador.

Posteriormente, Henricksen; Indulska e Rakotonirainy (2002) realizam uma classificação de contexto baseados nas suas associações, as quais podem ser estáticas ou dinâmicas. O contexto dinâmico tem características de persistência relacionadas à forma como as informações de contexto são obtidas. Essas características podem ser classificadas de acordo com a fonte: *sensed* (fornecidas por um sensor), *derived* (derivadas de outra fonte) ou *profiled* (definidas por uma pessoa ou por um serviço).

Considerando a aquisição e a persistência da informação, realizamos as seguintes classificações:

- percebida por sensores: informação gerada através dos sensores com alguma precisão;
- definida pelo usuário: o usuário fornece alguns valores à aplicação durante uma configuração. Valores padrão devem sempre estar disponíveis;
- fornecida por um serviço: A informação é dada por algum fornecedor de serviços, como uma agenda, na qual há uma lista de atividades da pessoa;
- derivada: A qual pode ser aprendida ou inferida.

Feng, Apers e Jonker (2004) apresentam uma categorização de contexto centrado na pessoa e no ambiente (Tabela 1).

Tabela 1 - Categorização de contexto centrado na pessoa e no ambiente

Centrado na pessoa	perfil da pessoa: interesse, hábitos, preferências
	comportamento da pessoa: intenções, atividades, tarefas.
	estado psicológico: temperatura, batimento cardíaco.
Centrado no ambiente	ambiente físico: tempo, localização, temperatura, iluminação.
	dispositivos: hardware, software, comunicação, nível de bateria.

2.2.4. Níveis de interação

Em níveis de interação, abordaremos a forma da interação entre a pessoa e o ambiente. De acordo com Chen e Kotz (2000), temos a definição de duas formas do uso do contexto ativa e passiva:

- ciente de contexto de forma ativa. A aplicação automaticamente se adapta a descoberta do contexto, mudando seu comportamento;
- ciente de contexto de forma passiva. A aplicação apresenta a informação nova e atualizada ao usuário, e a armazena para um uso posterior.

Posteriormente Barkhuus e Dey (2003) acrescentam uma terceira forma, a personalização. Os autores apresentam um estudo das duas formas de interação e sinalizam que o usuário perde o controle do ambiente usando aplicações cientes ao contexto de forma passiva ou ativa. A vantagem da personalização é que o ambiente atua autonomamente e o usuário recebe a informação ou o serviço solicitado no momento em que ele solicita e de forma apropriada.

Segundo Dey, Salber e Abowd (2001) existem três categorias de características que aplicações cientes ao contexto podem suportar. Acrescentemos a personalização a esta categoria:

- **apresentação de informação e serviços ao usuário:** Fornecer de forma apropriada informação e serviços que o usuário requeira. Nesta característica é conveniente ressaltar que o usuário tem que ser assistido e não deve ser interrompido ou incomodado com o serviço. Portanto, é conveniente considerar o perfil da pessoa, suas preferências e seu estado emocional;
- **execução automática dos serviços do usuário:** A execução dos serviços pode acontecer de forma automática. Como foi discutida anteriormente, a execução automática ou de forma ativa, faz com que o usuário perca o controle do ambiente. Esta execução tem que ser personalizada;
- **união de informações de contexto para um uso posterior:** Este tipo de característica é muito importante, pois aproveita a informação anterior juntamente com a situação atual, permitindo que o ambiente encontre relações na informação e forneça novos serviços baseados nos anteriores.

2.3. Web Semântica

2.3.1. Definição

A *World Wide Web* mudou a forma de comunicação com outros e a forma de como os negócios são conduzidos (ANTONIOU; HARMELEN, 2004). Nela existe uma grande quantidade de informações que são compreendidas somente por humanos, mas com falta de estrutura e expressividade para processamento pelas máquinas.

“A Web Semântica é uma extensão da web atual no qual o significado da informação está bem definido, permitindo um melhor trabalho cooperativo entre computadores e pessoas” (BERNERS-LEE; HENDLER; LASSILA, 2001).

A Web Semântica promete fornecer um modelo para conectar diferentes fontes de informações como páginas web, banco de dados e mesmo informações de nossa vida diária. Desse modo possibilita uma ampla variedade de aplicações e sistemas que se beneficiam da informação processada pelas máquinas (TJOA et al., 2005).

2.3.2. Ontologias

As ontologias são os blocos principais da Web Semântica para a representação de conhecimento. As ontologias provêm uma fácil interoperabilidade entre sistemas de informação, processada de forma inteligente por agentes e permite o reuso de conhecimento entre sistemas (PINTO; MARTINS, 2004).

Ontologia é um termo originário da filosofia usado para representar uma visão do mundo em um sistema de categorias. Uma das definições mais citadas na literatura é a de Gruber (1993) que define ontologia como:

“Uma ontologia é uma especificação formal, explícita e compartilhada de uma conceitualização”. Posteriormente Studer, Benjamins e Fensel (1998) analisam cada um dos termos desta definição:

- conceitualização: refere-se a um modelo abstrato de algum fenômeno no mundo, pela identificação de conceitos relevantes desse fenômeno;
- explícita: significa que o tipo de conceito usado e suas restrições estão explicitamente definidos;
- formal: refere-se ao fato de que a ontologia pode ser compreendida pelas máquinas;
- compartilhada: refere-se à noção de que uma ontologia captura um conhecimento aceito por um grupo de pessoas e não de forma individual.

2.3.3. Processo de construção de ontologias

A construção de ontologias é um processo que tem suas bases na engenharia de software. Para a construção de ontologias foram apresentadas diferentes metodologias, entre as quais cabe mencionar a METHONTOLOGY (LOPEZ et al., 1999), TOVE (GRÜNINGER, 1996) (GRÜNINGER; FOX, 1995), ENTERPRISE (USCHOLD; KING, 1995), e o Método de Desenvolvimento 101 (NOY; MCGUINNESS, 2001).

Neste trabalho foi escolhido como metodologia para a construção de ontologias o Método de Desenvolvimento 101 (NOY; MCGUINNESS, 2001), a qual caracteriza-se por ser simples e permitir a construção de ontologias desde o início ou mediante reuso de outras ontologias. Na sequência são descritos os setes passos citados no Método de Desenvolvimento 101, os quais são realizados em um processo iterativo.

1. determinar o domínio e o âmbito da ontologia;
2. considerar o reuso de outras ontologias;
3. enumerar importantes termos na ontologia;
4. definir as classes e hierarquia de classes;

5. definir as propriedades das classes;
6. definir a cardinalidade, tipo, domínio (*domain*) e valor (*range*) das propriedades;
7. criar as instâncias.

2.3.4. Linguagens da web semântica

A linguagem XML é a abreviação de *EXtensible Markup Language* (Linguagem extensível de formatação) (BRAY; PAOLI; SPERBERG-MCQUEEN, 2006) a qual é uma especificação técnica desenvolvida pela World Wide Web Consortium (W3C). O XML é uma linguagem que permite definir *tags*¹ personalizados sobre documentos. A linguagem XML é definida como o formato universal para dados estruturados na Web. A linguagem XML carece de expressividade e raciocínio.

O *Resource Description Framework* (RDF) (MANOLA; MILLER, 2004) é essencialmente um modelo de dados (ANTONIOU; HARMELEN; 2004). O RDF tem sua sintaxe em XML e herda todos os benefícios do XML. O elemento básico do RDF é o triplo: objeto, atributo (ou predicado) e valor, conhecido como sentença. As sentenças podem ser apresentadas em forma de grafo direcionado, no qual os nós são os sujeitos e os objetos e o predicado é o arco entre os nós.

O RDF permite descrever recursos, os quais são os objeto que queremos descrever. Cada recurso é identificado usando um *Uniform Resource Identifier* (URI). O RDF por si mesmo não fornece suficiente expressividade. O *RDF Schema* foi recomendado pela W3C para especificação de vocabulários RDF.

A linguagem OWL (SMITH; WELTY; MCGUINNESS, 2004) é o último padrão proposto pelo W3C para desenvolver linguagens adequadas para raciocinar sobre a semântica da informação na Web. O OWL pode ser usado para representar de forma explícita o significado de termos em vocabulários e suas relações entre os

¹ Estruturas de linguagem de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim.

termos. A representação dos termos e suas inter-relações são conhecidas como uma ontologia.

O OWL tem três espécies diferentes: OWL-Lite, OWL-DL, OWL-Full:

- OWL-Lite: É a mais simples, permite definir uma simples hierarquia de classes e simples restrições;
- OWL-DL: Tem um poder de expressividade maior que OWL-Lite e é baseado na Lógica Descritiva. É possível executar raciocínio automatizado em OWL-DL;
- OWL-Full: Tem maior expressividade, utilizada quando é mais importante a expressividade que garantir a decidibilidade. Não é possível executar raciocínio automatizado em OWL-Full.

2.3.5. Serviços Web Semânticos

Adotamos a definição de Serviço Web (*Web Service*) definido pelo W3C: Um Serviço web é um sistema de software desenhado para suportar interação interoperável máquina-máquina sobre uma rede. Ele tem uma interface descrita em um formato processável por máquina.

O W3C define Serviços Web como sistemas de software identificados por um URI e especificados por um documento de descrição baseada na linguagem XML. Os serviços web provêm uma forma padrão de interoperação entre diferentes aplicações software, executando em uma variedade de plataformas e/ou *frameworks*.

O problema dos serviços web é que carecem de uma semântica bem definida. Os SWS são serviços web anotados semânticamente. A descrição semântica dos Serviços Web inclui suas características, conteúdo e funcionalidades, com o intuito de descrever formalmente os Serviços Web em uma linguagem compreendida pelas máquinas.

McIlraith e Martin (2003) afirmam que os SWS permitiram uma nova faixa de tarefas executadas previamente pelas pessoas, incluindo composição, interoperação, monitoração e recuperação automatizada. Utilizando a semântica, o processo de descoberta tende a ser mais preciso que em uma descoberta pelo *matching* de simples palavras-chave (sintático).

A descoberta é o processo de encontrar Serviços Web com uma determinada característica. As abordagens de descoberta de serviços atualmente estão baseadas em um *matching* sintático, os quais não têm bom resultados (BROENS et al., 2004). Existem diferentes propostas para a realização da visão dos SWS; entre as que mais se ressaltam e também é recomendada pelo W3C, temos o OWL-S.

2.3.6. OWL-S

O OWL-S (MARTIN et al., 2004) é uma ontologia de alto nível, constituindo-se em continuação do projeto DAML-S (DAML-S, 2008), para anotar semânticamente Serviços Web. O OWL-S está construído acima do OWL.

Segundo Martin et al. (2004) a alta expressividade dada pelo OWL-S na descrição de Serviços Web, pode permitir uma automatização mais flexível da provisão e uso de serviços e suporte na construção de ferramentas e tecnologias mais poderosas.

O OWL-S fornece três tipos essenciais de conhecimento sobre um serviço (Ver Figura 4):

- um Perfil (*profile*) descreve as funcionalidades do serviço. O que o serviço faz;
- um Modelo (*model*) descreve detalhadamente como o serviço trabalha;
- um *Grounding* descreve como acessar o serviço.

Figura 4 - OWL-S Ontologia de serviços. Extraído de (MARTIN et al., 2004)

O perfil OWL-S foca em dois aspectos da funcionalidade do serviço que formam os IOPE (*inputs, outputs, preconditions, effects*):

- entradas (*inputs*) e saídas (*outputs*) representam a transformação da informação;
- condições (*preconditions*) e efeitos (*effects*) que representam a mudança de estado produzida pela execução do serviço.

É importante ressaltar que a ontologia *Profile* não fornece um esquema para descrever instâncias IOPE's; a ontologia *Process* define tal esquema. O modelo do serviço dá uma perspectiva mais detalhada de como o serviço opera e pode ser visto como um processo. Um processo em OWL-S tem duas funções “transformar” um conjunto um conjunto de entradas a um conjunto de saídas e uma transição no mundo de um estado a outro, essa transição é descrita pelas condições e efeitos do processo.

A classe *Process* tem três tipos de processos: atômicos (*atomic*), compostos (*composite*) e Simples (*simple*). Os processos atômicos são diretamente invocáveis e não tem sub-processos. Processos simples não são invocáveis e não são associados a um *grounding*. Processos compostos são divididos em outros

processos (não compostos e compostos), sua divisão pode ser especificada utilizando construtores de controle como: *Sequence*, *If-Then-Else*, *Split*, *Split-Joint* (MARTIN et al., 2004).

O *grounding* de um serviço especifica detalhes de como acessar um serviço. Um *grounding* pode ser visto como um mapeamento desde uma especificação abstrata a uma especificação concreta. O *grounding* define regras de mapeamento para o enlace de processos atômicos OWL-S a operações WSDL (*Web Services Description Language*) (CHRISTENSEN et al., 2001).

3. MODELO SEMÂNTICO DE CONTEXTO

Neste capítulo é descrito o OMC, o qual é um conjunto de ontologias para a modelagem de contexto em Ambientes Pervasivos.

3.1. Modelagem de contexto

Um modelo bem desenhado é a chave principal para qualquer sistema ciente de contexto (STRANG; LINNHOF-POPIEN, 2004). Uma boa representação do contexto deve permitir uma ampla faixa de possibilidades e uma real separação do sensoriamento do contexto e a reação programável ao contexto (ABOWD; MYNATT, 2000).

Segundo uma análise feita por Strang e Linnhoff-Popien (2004), destaca-se o uso de ontologias como a técnica mais promissora para a modelagem da informação de contexto. Sendo assim, serão utilizadas ontologias para a modelagem de contexto neste trabalho.

As ontologias são expressas em OWL, e as regras no SWRL. Para a modelagem dos serviços utiliza-se a ontologia de serviços OWL-S, que fornece os elementos necessários para descrever semanticamente um serviço e contém construtores para descrever fluxos de trabalho. Será utilizada essa capacidade para descrever o comportamento do Ambiente Pervasivo.

Neste trabalho foi escolhido como metodologia para a construção de ontologias o Método de Desenvolvimento 101 (NOY; MCGUINNESS, 2001).

3.2. Visão geral

Na Figura 5 são apresentadas as ontologias para a modelagem de contexto (OMC). A união entre as ontologias é realizada mediante um mecanismo de importação, sendo utilizadas setas direcionadas para representar graficamente que uma ontologia importa outra ontologia. Para importar outra ontologia foi utilizado o construtor *owl:imports*.

Figura 5 - Visão geral das ontologias do OMC

A modelagem de contexto tem que ser dividida em dois níveis (GU et al., 2004), (CHEN et al., 2004) (BULCÃO; PIMENTEL, 2005). A abordagem deste trabalho também considera uma divisão em dois níveis:

- no primeiro nível encontram-se as ontologias independentes de domínio, sendo elas: **Person**, **Space**, **Time**, **Device**, **ComputationalDevice**, **Activity**, **Preference**, **Policy** e **Service**;
- no segundo nível encontram-se as ontologias dependentes de domínio. Na Seção 4 será apresentado um Estudo de Caso cujo domínio é uma Casa Inteligente. As seguintes são as ontologias específicas para esse domínio: **HomePerson**, **HomeSpace**, **HomeDevice**, **HomeActivity**, e **OpenDoorService**.

Em um trabalho anterior (PONCE et al., 2006) apresenta-se uma visão inicial do modelo de contexto proposto (OMC). Neste modelo, são definidas ontologias para a modelagem de pessoas, espaço, dispositivos, tempo e políticas de privacidade. Posteriormente, foi definida a integração da modelagem de serviços em OWL-S e a modelagem de contexto utilizando OWL (PONCE et al., 2007). Esta integração apresenta a definição de parâmetros contextuais e a definição de fluxos de trabalhos dos serviços ou comportamento do Ambiente Pervasivo.

O modelo de contexto proposto neste trabalho (OMC) apresenta uma versão melhorada dos modelos anteriores (PONCE et al., 2006), (PONCE et al., 2007).

- As ontologias **Space**, **Device**, **Person**, e **Police** foram redefinidas, apresentam-se novas ontologias como **Activity** e **Preference** e as ontologias que são dependentes do domínio de uma Casa Inteligente.
- O modelo é definido em módulos e apresenta uma divisão em dois níveis;
- Na integração da modelagem de contexto e serviços são incluídas as propriedade **service:providedBy** e **service:provides**. Estas propriedades permitem descrever quem forneceu o serviço.

Na seqüência serão descritas detalhadamente as ontologias do OMC.

3.3. Pessoas

A modelagem de pessoas em um Ambiente Pervasivo é muito importante uma vez que a resposta do ambiente depende da pessoa. O Ambiente Pervasivo identifica a pessoa, possuindo informações relacionadas a esta pessoa, tais como seu nome, idade, sexo, etc. Na seqüência serão descritas as ontologias para a modelagem de pessoas.

3.3.1. Ontologia **Person**

Na construção da ontologia **Person** foram reutilizados os conceitos da ontologia *Friend of a Friend (Foaf)* (BRICKLEY; MILLER, 2007). A classe **Person**, define a pessoas em um Ambiente Pervasivo. A ontologia **Person** (Ver Figura 6) descreve o seguinte tipo de informação:

- dados pessoais que incluem o nome (**name**), idade (**age**), data de nascimento (**birthDate**), nacionalidade (**nationality**), entre outros;
- informações de contato de uma pessoa, como seu endereço (**address**), telefone (**phone**), entre outros;
- os grupos aos quais as pessoas pertencem.

Na Figura 6¹ se encontra uma visualização gráfica da ontologia **Person**. As classes são representadas com um marco retangular, as propriedades são representadas mediante um retângulo arredondado. Finalmente as instâncias são apresentadas com retângulo branco com borda. As classes **Person**, **Group** e **Sex** são definidas como *disjoint*. A classe **Person** é *disjoint* com a classe **Group** significa que elas são diferentes e não existe interseção entre as

¹ Figura gerada utilizando o *plugin* GrOWL (Veja Seção 5.1.2)

instancias de essas classes. Nenhuma instancia pode pertencer à classe **Person** e a classe **Group**. A notação ilustrada na Figura 6 será utilizada nas modelagens subseqüentes.

Figura 6 - Visualização gráfica da ontologia **Person**

3.3.2. Ontologia **HomePerson**

A ontologia **HomePerson** é uma ontologia específica do domínio da Casa Inteligente, que importa a ontologia **Person** e a estende com informação sobre as relações entre pessoas como “pai de” (**parentOf**), “esposo de” (**spouseOf**).

Esta ontologia faz reuso de conceitos da ontologia **Relationship** (DAVIS; VITIELLO, 2005). A ontologia **HomePerson** inclui também as definições de Pessoa Adulta, Pessoa Jovem, Pai, Homem, Mulher e Residente.

3.4. Espaço

Os primeiros trabalhos sobre a modelagem de contexto consideram amplamente a localização das pessoas, já que é possível fornecer serviços relativos a essa localização.

A modelagem de espaço inclui descrição dos espaços, suas formas, suas relações espaciais e suas coordenadas geográficas. Para a modelagem deste tipo de informação utiliza-se de conceitos de outras ontologias como **GeoOntology** (LING et al., 2006) (GEOONTOLOGIES ..., 2004) e wgs84_pos (BRICKLEY, 2003).

As ontologias para a modelagem de espaços são: **Space** e **HomeSpace**. As ontologias para a modelagem de outra informação relacionada a espaços são: **geoDescription**, **geoRelations** e **geoCoordinateSystem**.

3.4.1. Ontologia Space

A ontologia **Space** descreve espaços e a localização de objetos e pessoas. A classe **SpatialThing** é a classe principal da ontologia **Space** e representa objetos espaciais. Esta classe é formada por duas classes:

- **GeographicalObject** que representa um objeto geográfico. Dividem-se em **StaticThing** e **MovableThing**;
- **GeographicalSpace**, a qual se divide em espaços indoor e outdoor; Espaços indoor como uma sala (**room**), um andar (**floor**). Espaços outdoor como cidade (**city**), país (**country**) e continente (**continent**). Esta ontologia também permite a definição de entidades políticas.

A ontologia **Space**, permite definir a localização de pessoas e objetos, para o qual utiliza a propriedade **contains**. Esta propriedade **contains** é inversa à propriedade

locatedIn. Na Figura 7 é apresentada uma visualização gráfica da ontologia **Space**, nesta figura se detalha a classe **GeographicalSpace**.

Figura 7 - Visualização gráfica da ontologia **Space**

3.4.2. Ontologia **HomeSpace**

A ontologia **HomeSpace** importas as ontologias **Space** e **HomePerson**. A ontologia **HomeSpace** é uma ontologia específica do domínio da Casa Inteligente. A ontologia **HomeSpace** estende a informação da ontologia **Space** com informação sobre os espaços de uma Casa Inteligente.

3.4.3. Ontologia **geoDescription**

A ontologia **geoDescription** permite a definição de objetos espaciais, esta ontologia importa a ontologia **Space** e permite representar a forma que o objeto espacial tem.

Esta ontologia se relaciona com a ontologia **Space** mediante a propriedade *shape*, na qual um objeto espacial **SpatialThing** tem uma descrição espacial **spatialDescription**. Esta descrição é definida mediante a classe **GeometricFeature**, que define figuras geométricas para a representação da forma de um objeto espacial. Estas formas são descritas mediante coordenadas “xy”, ou “xyz” e é do tipo *String*. Por exemplo, a definição da forma de um quadrado de lado 10 x 10 é assim: “0.0, 0.0 10.0, 0.0 10.0, 10.0 0.0, 10.0”.

3.4.4. Ontologia **geoRelations**

Esta ontologia permite a definição de relações entre dois objetos espaciais. Essas relações podem ser qualitativas e quantitativas. A ontologia **geoRelations** permite definir as seguintes relações espaciais:

- relações topológicas: As relações topológicas referem-se às propriedades como conectividade, adjacência e intersecção entre objetos espaciais.
- relações cardinais: Utiliza-se uma direção de cardinalidade (Leste, Oeste, Norte, Sul) para descrever a localização de um objeto em relação ao outro. Também permite definir relações relativas de direção (*esquerda, direita, acima, abaixo*);
- relações de proximidade: Esta relação define uma noção de proximidade entre dois objetos. A proximidade pode ser qualitativa (perto, longe) ou quantitativa na qual se utiliza unidades espaciais como o metro;
- relações mereológicas: Uma relação mereológica estabelece que um objeto é “parte de” ou “tudo de”.

Na Figura 8 é apresentada uma visualização gráfica da ontologia **geoRelations** na qual apresenta-se as principais classes e instâncias.

Figura 8 - Visualização gráfica da ontologia **geoRelations**

3.4.5. Ontologia **geoCoordinateSystem**

Um objeto espacial (**SpatialThing**) é localizado utilizando um sistema de coordenadas. A classe **CoordinateSystem** descreve o sistema de coordenadas de um objeto espacial e deriva-se em duas classes **GeographicCoordinateSystem** e **LocalCoordinateSystem**.

A primeira classe representa um sistema de coordenada geográfica; um dos sistemas de coordenadas geográficos mais utilizado é o WGS84Coordinates, o qual descreve a localização utilizando a latitude, longitude e altitude. A segunda classe (**LocalCoordinateSystem**), define uma classe de sistemas de coordenadas específicas para um objeto ou lugar.

3.5. Dispositivos

Em um documento público do projeto *Amigo*, D3.3 [...] (2006) foram detalhadas ontologias para dispositivos que pertencem aos domínios de ambiente domésticos, de eletrônicos de consumo, e ambientes de computação pessoal (PC).

A modelagem de dispositivos utiliza as ontologias: **Device**, **ComputationalDevice** e **HomeDevice**. As duas últimas reutilizam os conceitos e definições das ontologias do Projeto *Amigo*.

3.5.1. Ontologia **Device**

A Descrição das propriedades gerais dos dispositivos é dada na ontologia **Device** (ver Figura 9), ela descreve quem é o proprietário do dispositivo, o estado e as operações do dispositivo.

Figura 9 - Visualização gráfica da ontologia **Device**

3.5.2. Ontologia **ComputationalDevice**

A ontologia **ComputationalDevice** (ver Figura 10) descreve os dispositivos computacionais em um Ambiente Pervasivo. A classe **ComputationalDevice** é subclasse da classe **Device** e descreve um dispositivo computacional como: PDA (*Personal Digital Assistant*), Laptop e PC (*Personal Computer*).

A propriedade **hardwareComponent** descreve os componentes hardware de um dispositivo, que são descritos pela classe **HardwarePlatform**. Esta última compreende memória, bateria, placa de vídeo, Central Processing Unit (CPU) e armazenamento. A propriedade **softwareComponent** descreve as aplicações e o software dos dispositivos.

A classe **Capability** descreve as capacidades de comunicação dos dispositivos, que são: texto, vídeo e áudio. A classe **Peripheral** descreve os periféricos, os quais são classificados em **Duplex**, **Input** e **Output**.

Figura 10 - Visualização gráfica da ontologia **ComputationalDevice**

A ontologia **ComputationalDevice** apresenta o vocabulário para descrever conceitos de um dispositivo computacional, dependente do domínio da aplicação, a qual pode ser estendida de acordo com as necessidades. exemplo, descrição da frequência do CPU e software descrição do Sistema Operacional (SO) do dispositivo.

3.5.3. Ontologia **HomeDevice**

A ontologia **HomeDevice** permite descrever dispositivos de uma Casa Inteligente. Os dispositivos são classificados em sensores (**sensors**), atuadores (**actuators**) e eletroeletrônicos (**appliances**).

- os sensores podem detectar mudanças no ambiente;
- os atuadores interagem com o ambiente;
- os eletroeletrônicos os quais podem ser programados.

Na Figura 11 é apresentada a hierarquia de classes dos dispositivos de uma casa inteligente.

Figura 11 - Visualização gráfica da ontologia **HomeDevice**

3.6. Tempo

A modelagem de tempo permite descrever instantes e intervalos de tempo, em que acontecem as coisas, com suas respectivas durações. A ontologia **Time** é o OWL-Time (HOBBS; PAN, 2006) formalmente conhecido como DAML-Time (não foi feita alteração nenhuma na ontologia). O OWL-Time é utilizado para descrever conceitos temporais, conteúdo temporal de páginas web e propriedades temporais de serviços web. O OWL-Time foi utilizada para descrever o tempo em Ambientes Pervasivos no *Context Broker Architecture* (CoBra) (CHEN; FININ; JOSHI, 2004) e o *Semantic Context Model* (SecoM) (BULCÃO; PIMENTEL, 2005).

3.6.1. Ontologia **Time**

A ontologia **Time** permite representar o seguinte tipo de conhecimento:

- descrição de entidades temporais: A classe **TemporalEntity** representa uma entidade de tempo que pode ser um instante ou um intervalo;
- relações topológicas temporais: Define relações topológicas temporais como quando dois intervalos são iguais (**intervalEquals**), quando um intervalo acontece primeiro que o outro;
- descrição de duração: Para definir a duração se utiliza a propriedade **durationOf**, cujos argumentos são: ano, mês, semana, dias, horas, minutos e segundos;
- zona de tempo: Um instante de tempo está associado a uma zona de tempo;
- descrição de uma data e tempo: Uma descrição de uma data e tempo tem as seguintes propriedades: unidade de tempo (**unitType**), seu valor (**year**, **month**, **week**, **day**, etc.) e zona de tempo (**timeZone**);

3.7. Atividades

A modelagem de atividades inclui a descrição de atividades, o tipo de atividade, em que tempo ocorre, o lugar e os atores que a realizam.

3.7.1. Ontologia **Activity**

A ontologia **Activity** permite a modelagem de eventos, atividades em um ambiente pervasivo. A ontologia **Activity** importa as ontologias **Person**, **Space** e **Time**.

Na Figura 12 é apresentada uma visualização gráfica das classes, propriedades e instâncias da ontologia **Activity**. Cada atividade tem associado um evento, e cada evento é descrito por seu nome e tipo. As atividades são descritas com prioridades qualitativas ou quantitativas. Uma atividade tem como participante uma pessoa, isto é definido utilizando a propriedade **hasParticipant**.

Figura 12 - Visualização gráfica da ontologia **Activity**

As atividades são definidas como eventos espaço-temporais. Esta acontece em um espaço, a propriedade **place** define esse espaço geográfico **geographicalSpace**; e é dada em um determinado tempo, para qual se utiliza a classe **TemporalEntity** da ontologia **Time**.

3.7.2. Ontologia **HomeActivity**

A ontologia **HomeActivity** permite descrever as atividades de uma Casa Inteligente. Por exemplo, a atividade “Ricardo assiste televisão na sala”. Tem como participante Ricardo, o local da atividade é na sala e tem associado o evento “assistir televisão” cujo tipo é de entretenimento. A ontologia **HomeActivity** importa as ontologia **Activity** e **HomeSpace**

3.8. Políticas de privacidade

As políticas de privacidade permitem estabelecer regras ou políticas de controle ao acesso aos serviços e à informação. Por exemplo, definir quem pode acessar a informação pessoal como informação das compras que uma pessoa realiza hoje, que pessoas podem utilizar um determinado dispositivo, quais são as pessoas que podem ingressar em um espaço.

3.8.1. Ontologia **Police**

A ontologia **Police** permite definir políticas de privacidade em um Ambiente Pervasivo. A ontologia **Police** importa as ontologias **Person**, **Device** e **Activity**.

A principal classe desta ontologia é **Police**, cujas propriedades são: nome, proprietário, data de criação e validade. Uma política de privacidade estabelece quais são as pessoas autorizadas a realizar uma determinada ação. Esta ação é definida como o “**target**”. Entre as ações controladas foram definidas ações dos dispositivos e os eventos. As políticas de privacidade são representadas utilizando regras codificadas no SWRL.

3.9. Preferências

A resposta do ambiente depende das preferências da pessoa. Um serviço personalizado permite que o ambiente forneça os serviços que a pessoa deseja. A informação do perfil da pessoa possibilita uma interação personalizada em diferentes ambientes. Deve existir um mecanismo para ir armazenando e compartilhando as preferências da pessoa.

3.9.1. Ontologia **Preference**

A ontologia **Preference** permite definir preferências das pessoas, as quais têm uma descrição quantitativa e qualitativa. As preferências são divididas em preferências por atividades, por aplicações, por serviços e por dispositivos.

As preferências têm uma característica diferente das demais informações de contexto, devido ao fato de terem um caráter dinâmico, que varia de pessoa para pessoa e de acordo a situação. Utilizaremos regras codificadas em SWRL para expressar preferências.

3.10. Serviços cientes de contexto

A modelagem de serviços utiliza a modelagem de contexto descrita anteriormente. É utilizada a ontologia de serviços OWL-S para definir serviços cientes de contexto. A ontologia OWL-S é uma ontologia escrita em OWL, portanto é possível a integração da modelagem de serviços e informação de contexto.

Em um trabalho anterior alguns autores deste trabalho (PONCE et al., 2007) apresentaram a integração da modelagem de serviços em OWL-S e a modelagem de contexto utilizando OWL. Esta integração apresenta a definição de parâmetros contextuais e a definição de fluxos de trabalhos dos serviços ou comportamento do Ambiente Pervasivo.

O presente trabalho é continuação do trabalho anterior (PONCE et al., 2007), no qual são descritos os serviços fornecidos pelos dispositivos. Na integração da modelagem de contexto e serviços são incluídas as propriedades ***service:providedBy*** e *service:provides*. Estas propriedades permitem descrever quem forneceu o serviço.

4. ESTUDO DE CASO

No capítulo anterior (Capítulo 3), foram apresentadas as ontologias para a modelagem de contexto em um Ambiente Pervasivo cujo domínio de aplicação é uma Casa Inteligente. Na seqüência é descrito e desenvolvido o Estudo de Caso.

No Apêndice A, é descrito brevemente um roteiro para a elaboração de um Estudo de Caso.

4.1. Enunciado

O enunciado do Estudo de Caso foi dividido em duas partes, ingressar na casa e configuração.

4.1.1. Ingressar na casa

Rodrigo vai para a casa do seu amigo Carlos, (onde o sistema só permite o ingresso de pessoas com autorização. Caso não exista tal autorização, a pessoa pode solicitá-la usando o interfone).

Chegando à casa do amigo e perto da porta, o sistema o identifica como sendo o Rodrigo e passa a consultar se ele tem autorização de ingresso. A consulta retorna que Rodrigo não tem autorização de ingresso.

Utilizando o interfone, ele solicita autorização a Carlos que a concede. Agora, o sistema lhe permite ingressar na casa a partir de uma configuração prévia do seu perfil como convidado.

4.1.2. Configuração do ambiente

O sistema abre a porta enviando uma mensagem de boas vindas, também obtém informações pessoais do Rodrigo, as quais estão armazenadas no seu *PDA*.

Rodrigo ingressa na sala e senta-se em frente à TV. O sistema informa que ele provavelmente tem interesse em assistir TV. Esta dedução é dada já que a atividade dele é de descanso; e seu perfil indica que assistir TV é uma das atividades favoritas dele. O perfil também indica o tipo de programação que Rodrigo assiste e liga a TV no canal que apresenta a programação de sua preferência, considerando aspectos de ambientação como o som da TV e a iluminação do ambiente.

4.2. Descrição dos serviços

A seguir, são descritos os serviços ingressar na casa e configuração implementados segundo o Estudo de Caso.

4.2.1. Ingressar na casa

Quando o ambiente identifica a presença de uma pessoa, o programa lhe permite ou nega o acesso à casa. Esta atividade é composta pelos serviços: “Abrir Porta” (***OpenDoor***), “Pedir Permissão” (***RequestPermission***) e “Obter a informação da pessoa” (***GetPersonInformation***).

- serviço “Abrir Porta”, este serviço permite o acesso ou não à casa;
- serviço “Pedir Permissão”, este serviço estabelece comunicação com um residente da casa para solicitar autorização à entrada na mesma;
- serviço “Obter a informação da pessoa”, este último serviço está ligado ao serviço “Pedir Permissão” proporcionando a informação da localização dos

residentes da casa através de dispositivos móveis conectados ao sistema (celulares, PDA, etc.).

Na Figura 13, é apresentado um diagrama de atividades do serviço “Ingressar na casa”. A classe “Contexto”, representa a informação de contexto, como a presença de uma nova pessoa. A classe “Serviços”, representa os serviços da Casa Inteligente, como a identificação de pessoas, verificar autorização, solicitar permissão, abrir porta, entre outros. A classe “Políticas” e “Preferências”, representam as políticas de privacidade e preferências respectivamente.

Figura 13 - Diagrama de atividades do serviço “Ingressar na casa”

O serviço começa (Figura 13) quando uma nova pessoa é identificada, o ambiente realiza a identificação da pessoa, e verifica a autorização da pessoa a ingressar na

casa, isto é definido nas políticas de privacidade. Se a pessoa não for autorizada, ela pode pedir autorização. Quando a pessoa identificada tem autorização o ambiente obtém o perfil de preferências e posteriormente abre a porta.

4.2.2. Configuração do ambiente

Aqui se estabelece os parâmetros de configuração de acordo com a informação do contexto, considerando as preferências da pessoa e as políticas de privacidade. Esta atividade é composta pelos serviços "Obter Perfil da pessoa" (***getPersonPerfil***), "Configurar Casa" (***setHomeEnvironment***) e "Entretenimento por Televisão" (***TVEntertainment***).

- serviço "obter perfil da pessoa", este serviço obtém o perfil da pessoa que está armazenado no seu dispositivo pessoal;
- serviço "Configurar Casa", este serviço configura a casa de acordo com as preferências e as políticas de privacidade designadas para essa pessoa;
- serviço "Entretenimento por Televisão", este serviço liga a TV de forma automática considerando as preferências da pessoa.

4.3. Modelagem de serviços cientes de contexto

Em um trabalho anterior, Ponce et al. (2007) apresentam a modelagem de serviços cientes de contexto. Esta abordagem descreve a utilização de parâmetros contextuais e a modelagem do comportamento do ambiente como fluxo de trabalho do serviço. A seguir, será descrito o serviço "Abrir Porta".

4.3.1. Modelagem do serviço

O Serviço “Abrir Porta” (**OpenDoor**), permite o acesso de uma pessoa autorizada à casa. Na modelagem deste serviço utiliza-se o OWL-S 1.2 Para modelagem deste serviço são importadas as ontologias **HomeActivity**, **Police** e **Preference**.

O Quadro 1 mostra o serviço “Abrir Porta”, o qual é descrito em OWL-S. O serviço é descrito com um perfil (linha 1), um processo (linha 2) e um *grounding* (linha 3). Como foi apresentada na seção 3.10, é utilizada a propriedade **service:provide** para especificar quem fornece o serviço. Neste caso ele é fornecido pelo dispositivo “Porta” (**DoorLock_1**) (linha 4), que é a porta de acesso à casa.

```

0  <service:Service rdf:ID="SVC_OpenDoor">
1    <service:presents rdf:resource="#Profile_OpenDoor"/>
2    <service:describedBy rdf:resource="#AuthorizePerson"/>
3    <service:supports rdf:resource="#WsdGrounding_OpenDoor"/>
4    <service:providedBy rdf:resource="#homDev_1;DoorLock_1"/>
5  </service:Service>

```

Quadro 1 - Descrição de um serviço em OWL-S

O Quadro 2 mostra uma descrição do dispositivo “Porta”, o qual é responsável pela ação de abrir a porta de acesso à casa. Utilizando a propriedade **service:provides** especifica-se que este dispositivo fornece o serviço “Abrir Porta”.

```

0  <rdf:Description rdf:about="#homDev_1;DoorLock_1">
1    <service:provides rdf:resource="#SVC_OpenDoor"/>
2  </rdf:Description>

```

Quadro 2 - Descrição do dispositivo “Porta”

4.3.2. Parâmetros contextuais

A ontologia de perfil não fornece um esquema para descrever instâncias IOPE (*inputs, outputs, preconditions, effects*). O esquema é definido na ontologia do

processo e utilizado pela ontologia do perfil. As instâncias IOPE são utilizadas como parâmetros contextuais (PONCE et al., 2007). Na Tabela 2 são descritos os parâmetros contextuais do Serviço “Abrir Porta”.

Tabela 2 - Parâmetros contextuais do Serviço “Abrir Porta”

	Parâmetro contextual	Descrição
<i>Entradas</i>	<i>Person_EnterHome</i>	<i>Quem quer entrar na casa</i>
	<i>Space_EnterHome</i>	<i>A que espaço pretende</i>
	<i>Time_EnterHome</i>	<i>O tempo que vai ser o acesso</i>
<i>Saídas</i>	<i>Message</i>	<i>Mensagem que indica a ação efetuada</i>
<i>Precondições</i>	<i>DoorIsClosed</i>	<i>A porta tem que estar fechada</i>
	<i>PersonWantsEnterHome</i>	<i>Uma pessoa quer entrar na casa</i>
<i>Efeitos</i>	<i>OpenTheDoor</i>	<i>A porta é aberta</i>

O Quadro 3 mostra os parâmetros contextuais utilizados no perfil do serviço “Abrir Porta”.

```

0  <profile:serviceName rdf:datatype="&xsd:string"
1    >Profile_OpenDoor</profile:serviceName>
2    <profile:textDescription rdf:datatype="&xsd:string"
3    >Profile_OpenDoor</profile:textDescription>
4    <service:presentedBy rdf:resource="#SVC_OpenDoor"/>
5    <profile:hasInput rdf:resource="#Space_EnterHome"/>
6    <profile:hasInput rdf:resource="#Person_EnterHome"/>
7    <profile:hasInput rdf:resource="#Time_EnterHome"/>
8    <profile:hasOutput rdf:resource="#Message"/>
9    <profile:hasPrecondition rdf:resource="#PersonWantsEnterHome"/>
10   <profile:hasPrecondition rdf:resource="#DoorIsClosed"/>
11   <profile:hasResult rdf:resource="#Result_OpenTheDoor"/>
12 </profile:Profile>

13 <process:Result rdf:ID="Result_OpenTheDoor">
14   <process:hasEffect rdf:resource="#OpenTheDoor"/>
15 </process:Result>

```

Quadro 3 - Descrição dos IOPEs do perfil do serviço “Abrir Porta”

O Quadro 4 mostra os parâmetros contextuais do serviço “Abrir Porta”. O parâmetro da entrada ***Person_EnterHome*** é do tipo ***Person*** (linhas 0-2), ou uma pessoa, a qual é representada mediante a ontologia ***Person*** (Ver seção 3.3).

```

0 <process:Input rdf:ID="Person_EnterHome">
1   <process:parameterType rdf:datatype="&xsd:anyURI"
2     "&per;Person">/process:parameterType>
3 </process:Input>

4 <process:Input rdf:ID="Space_EnterHome">
5   <process:parameterType rdf:datatype="&xsd:anyURI"
6     "&spc;Indoor">/process:parameterType>
7 </process:Input>

8 <process:Input rdf:ID="Time_EnterHome">
9   <process:parameterType rdf:datatype="&xsd:anyURI"
10    "&tme;TemporalEntity">/process:parameterType>
11 </process:Input>

12 <process:Output rdf:ID="Message">
13   <process:parameterType rdf:datatype="&xsd:anyURI"
14     "&xsd:string">/process:parameterType>
15 </process:Output>

```

Quadro 4 - Definição dos IOPEs do processo do serviço “Abrir Porta”

4.3.3. Fluxo de trabalho do serviço

Em OWL-S é possível definir processos simples, atômicos e compostos. Um processo composto utiliza construtores de controle como: *Sequence*, *If-Then-Else*, *Split*, *Split-Join*, *While*, *Choice* entre outros. Neste trabalho, utilizam-se fluxos de trabalhos (*service workflows*) para descrever o comportamento do Ambiente Pervasivo (PONCE et al., 2007).

Os processos que compõem “Autorizar Pessoa” são: “Verificar parâmetros contextuais” e “Enviar mensagem”. Na Figura 14 é apresentado o Fluxo do trabalho do processo “Autorizar pessoa”. A descrição do processo “Autorizar pessoa” em OWL-S, é apresentada no Quadro 5, o qual é uma seqüência de dois processos: “Verificar parâmetros contextuais” (***CheckContextParameters***) e “Enviar mensagem” e (***SendMessage***).

Figura 14 - Fluxo de Trabalho do processo "Autorizar Pessoa"

```

0 <process:CompositeProcess rdf:ID="AuthorizePerson">
1   <process:name
2     rdf:datatype="&xsd:string">AuthorizePerson</process:name>
3   <process:composedOf rdf:resource="#Sequence_AuthorizePerson"/>
4 </process:CompositeProcess>
5 <process:Sequence rdf:ID="Sequence_AuthorizePerson">
6   <process:components
7     rdf:resource="#ControlConstructList_CheckContextParameters"/>
8 </process:Sequence>
9 <process:ControlConstructList
10   rdf:ID="ControlConstructList_CheckContextParameters">
11   <list:first rdf:resource="#Perform_CheckContextParameters"/>
12   <list:rest rdf:resource="#ControlConstructList_SendMessage"/>
13 </process:ControlConstructList>
14 <process:Perform rdf:ID="Perform_CheckContextParameters">
15   <process:process rdf:resource="#CheckContextParameters"/>
16 </process:Perform>
17 <process:ControlConstructList
18   rdf:ID="ControlConstructList_SendMessage">
19   <list:first rdf:resource="#Perform_SendMessage"/>
20   <list:rest rdf:resource="&list:nil"/>
21 </process:ControlConstructList>
22 <process:Perform rdf:ID="Perform_SendMessage">
23   <process:process rdf:resource="#SendMessage"/>
24 </process:Perform>
  
```

Quadro 5 - Parâmetros contextuais do processo composto "Autorizar Pessoa"

4.3.3.1. Processo “Verificar parâmetros contextuais”

O processo composto “Verificar parâmetros contextuais” é o núcleo do Serviço **OpenDoor**. No Quadro 6 é apresentada a descrição em OWL-S desse processo. Esse processo é composto por três processos atômicos (linhas 5-15). O construtor *Split-join* permite especificar processos executados concomitantemente. Na Figura 15 é apresentado o fluxo do trabalho do processo “Verificar parâmetros contextuais”.

```

0 <process:CompositeProcess rdf:ID="CheckContextParameters">
1   <process:hasInput rdf:resource="#personAllowed_EnterHome"/>
2   <process:hasInput rdf:resource="#AccessInstant"/>
3   <process:hasInput rdf:resource="#AllowedInterval"/>
4   <process:hasInput rdf:resource="#AllowedSpace"/>
5   <process:composedOf>
6     <process:Split-Join>
7     <process:components>
8       <process:ControlConstructBag>
9         <process:AtomicProcess rdf:about="#CheckPerson"/>
10        <process:AtomicProcess rdf:about="#CheckSpace"/>
11        <process:AtomicProcess rdf:about="#CheckTime"/>
12      </process:ControlConstructBag>
13    </process:components>
14  </process:Split-Join>
15 </process:composedOf>
16 </process:CompositeProcess>

```

Quadro 6 - O processo composto “Obter parâmetros contextuais”

Figura 15 - Fluxo de Trabalho do processo “Verificar parâmetros contextuais”

4.3.3.2. Processo “Enviar mensagem”

O processo atômico “Enviar Mensagem” é um processo com alguma mensagem como saída. Estas mensagens podem ser fornecidas por outro serviço através de um dispositivo, como um auto-falante que apresenta como mensagem de saída um sinal de áudio. Os resultados do processo podem ser:

- a pessoa tem autorização de ingressar. Neste caso se envia uma mensagem de boas vindas à pessoa e a porta é aberta automaticamente;
- a pessoa não consegue autorização de ingresso, mas ela pode solicitar autorização a um residente. Neste caso a mensagem é de invocar o Serviço “Solicitar permissão”;
- quando o resultado do serviço “Solicitar permissão” for negativo. Neste caso é enviada uma mensagem informando cordialmente que o pedido de ingresso à casa foi recusado.

4.4. Modelagem do contexto

4.4.1. Pessoas

Para a modelagem de pessoas foi utilizada a ontologia **Person**. No Quadro 7, apresenta-se parte de código em OWL da ontologia **Person**. Esta ontologia contém três classes **Person** (linhas 10-17), **Sex** e **Group**. Algumas das suas propriedades são idade (linhas 18-21), telefone (linhas 22-25), nome (linhas 26-29) e sexo (linhas 30-33). Para a modelagem dos relacionamentos entre pessoas foi utilizada a ontologia **HomePerson**. Esta ontologia importa a ontologia **Person**.

```

0  <?xml version="1.0"?>
1  <rdf:RDF
2    xmlns="http://www.pad.lsi.usp.br/context/person.owl#"
3    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
5    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
6    xmlns:owl11="http://www.w3.org/2006/12/owl11#"
7    xmlns:owl11xml="http://www.w3.org/2006/12/owl11-xml#"
8    xmlns:owl="http://www.w3.org/2002/07/owl#"
9    xml:base="http://www.pad.lsi.usp.br/context/person.owl">

10  <owl:Class rdf:ID="Person">
11    <owl:disjointWith>
12      <owl:Class rdf:ID="Sex"/>
13    </owl:disjointWith>

14    <owl:disjointWith>
15      <owl:Class rdf:ID="Group"/>
16    </owl:disjointWith>

17  </owl:Class>
18  <owl:DatatypeProperty rdf:ID="age">
19    <rdfs:range rdf:resource="&xsd:int"/>
20    <rdfs:domain rdf:resource="#Person"/>
21  </owl:DatatypeProperty>
22  <owl:DatatypeProperty rdf:ID="phone">
23    <rdfs:range rdf:resource="&xsd:string"/>
24    <rdfs:domain rdf:resource="#Person"/>
25  </owl:DatatypeProperty>
26  <owl:DatatypeProperty rdf:ID="name">
27    <rdfs:range rdf:resource="&xsd:string"/>
28    <rdfs:domain rdf:resource="#Person"/>
29  </owl:DatatypeProperty>

30  <owl:ObjectProperty rdf:about="#hasSex">
31    <rdf:type rdf:resource="&owl:FunctionalProperty"/>
32    <rdfs:range rdf:resource="#Sex"/>
33  </owl:ObjectProperty>

```

Quadro 7 - Parte de código da ontologia **Person**

No Quadro 8 apresenta-se parte de código em OWL da ontologia **HomePerson**. A propriedade “filho de” **childOf** (linhas 0-4) tem como domínio e valor desta propriedade a classe **Person**, sendo essa propriedade uma sub-propriedade da propriedade “conhece” **knows**. A propriedade “pai de” **parentOf** (linhas 5-10) é inversa com a propriedade “filho de”.

```

0  <owl:ObjectProperty rdf:about="#childOf">
1    <rdfs:subPropertyOf rdf:resource="#knows"/>
2    <rdfs:range rdf:resource="#person;Person"/>
3    <rdfs:domain rdf:resource="#person;Person"/>
4  </owl:ObjectProperty>

5  <owl:ObjectProperty rdf:about="#parentOf">
6    <owl:inverseOf rdf:resource="#childOf"/>
7    <rdfs:subPropertyOf rdf:resource="#knows"/>
8    <rdfs:domain rdf:resource="#person;Person"/>
9    <rdfs:range rdf:resource="#person;Person"/>
10 </owl:ObjectProperty>

```

Quadro 8 - Propriedades de relacionamentos da ontologia **HomePerson**

No Quadro 9 é apresentada a modelagem em OWL dos relacionamentos de Ricardo, a qual utiliza a ontologia **HomePerson**. Foram feitas as seguintes definições sobre a pessoa "**Person_1**" (linhas 4-9):

- seu nome é Ricardo;
- sua idade é 40 anos;
- esposo de Mariela (**Person_2**);
- pai de Alex (**Person_3**);
- trabalha com Rodrigo (**Person_5**);
- seu sexo é masculino.

```

0  <per:Person rdf:ID="Person_1">
1    <per:name rdf:datatype="&xsd:string">Ricardo</per:name>
2    <per:age rdf:datatype="&xsd:int">40</per:age>
3    <homePer:spouseOf rdf:resource="#Person_2"/>
4    <homePer:parentOf rdf:resource="#Person_3"/>
5    <homePer:worksWith rdf:resource="#Person_5"/>
6    <homePer:hasSex rdf:resource="#per;#MaleSex"/>
7  </per:Person>

```

Quadro 9 - Modelagem em OWL dos relacionamentos de Ricardo

A ontologia **HomePerson** possui a declaração de pessoa adulta, pessoa jovem, homem, mulher e pai. No Quadro 10 apresenta-se a descrição em OWL da classe pessoa adulta (**AdultPerson**).

A classe “pessoa adulta” é descrita como uma pessoa (linhas 1-14) cuja idade mínima é 18 anos (linhas 5-14). Para definir isto em OWL utiliza-se a definição de intersecção de uma pessoa (linhas 3-4) e alguém que satisfaça a seguinte restrição existencial (**someValuesFrom**) definido na propriedade idade da pessoa: A restrição (linhas 5-6) especifica que o valor da propriedade “idade” deve ser no mínimo um valor inteiro igual ou maior a 18 (linhas 10-14).

Também é definido que a classe “pessoa adulta” é complemento da classe “pessoa jovem” (linhas 21-25).

```

0  <owl:Class rdf:about="#AdultPerson">
1      <owl:equivalentClass>
2          <owl:Class>
3              <owl:intersectionOf rdf:parseType="Collection">
4                  <rdf:Description rdf:about="&person;Person"/>
5
6                  <owl:Restriction>
7                      <owl:onProperty rdf:resource="&person;age"/>
8                      <owl:someValuesFrom>
9                          <rdf:Description>
10                              <rdf:type rdf:resource="&owl;DataRange"/>
11                              <owl11:minInclusive
12                                  rdf:datatype="&xsd;int">18
13                              </owl11:minInclusive>
14                              <owl11:onDataRange
15                                  rdf:resource="&xsd;int"/>
16                              </rdf:Description>
17                          </owl:someValuesFrom>
18                      </owl:Restriction>
19
20              </owl:intersectionOf>
21          </owl:Class>
22      </owl:equivalentClass>
23      <rdfs:subClassOf>
24          <owl:Class>
25              <owl:complementOf rdf:resource="#YoungPerson"/>c
26          </owl:Class>
27      </rdfs:subClassOf>
28  </owl:Class>

```

Quadro 10 - Definição da classe “pessoa adulta”

4.4.2. Espaço

Na modelagem de espaços são modelados espaços e relações espaciais de uma Casa (Casa Inteligente), para o qual são utilizadas as ontologias ***Space, HomeSpace, geoDescription, geoRelation*** e ***geoCoordinates***.

A localização espacial dos objetos contidos em uma cozinha é apresentada no mapa da Figura 16. Neste mapa apresenta-se o espaço cozinha, a localização da pessoa (***Person_2***) e a mesa (***table***). A descrição da forma dos objetos é realizada através de uma figura geométrica (Seção 3.4.2).

Figura 16 - Mapa de um espaço geográfico da Casa Inteligente

No Quadro 11, apresenta-se a definição de objetos espaciais e sua forma. A cozinha (**Kitchen**) contém dois objetos espaciais “**Person_2**” e “**table**” (linhas 0-4) e sua forma é definida através das coordenadas “xy” de um polígono (linhas 5-9).

A seguir, são descritas as relações espaciais entre a pessoa (**Person_2**) e a mesa (**table**) definidos no Quadro 11.

```

0  <spc:Room rdf:ID="Kitchen">
1    <geoD:shape rdf:resource="#KitchenShape"/>
2    <spc:contais rdf:resource="#Table"/>
3    <spc:contains rdf:resource="#Person_2"/>
4  </spc:Room>
5  <geoD:MultiPolygon rdf:ID="KitchenShape">
6    <geoD:xyCoordinates rdf:datatype="&xsd:string"
7      >44.0,50.0 44.0,316.0
8      394.0,316.0 394.0,50.0 44.0,50.0</geoD:xyCoordinates>
9  </geoD:MultiPolygon>
10 <spc:Person rdf:ID="Person_2">
11   <geoD:shape rdf:resource="#Person_2Shape"/>
12 </spc:Person>
13 <geoD:MultiPolygon rdf:ID="Person_2Shape">
14   <geoD:xyCoordinates rdf:datatype="&xsd:string"
15     >354.0,228.0 354.0,369.0
16     480.0,369.0 480.0,228.0 354.0,228.0</geoD:xyCoordinates>
17 </geoD:MultiPolygon>
18 <geoD:MultiPolygon rdf:ID="TableShape">
19   <geoD:xyCoordinates rdf:datatype="&xsd:string"
20     >111.0,140.0 111.0,398.0 316.0,398.0
21     316.0,140.0 111.0,140.0 </geoD:xyCoordinates>
22 </geoD:MultiPolygon>

```

Quadro 11 - Descrição dos objetos contidos na cozinha

A pessoa tem três tipos de relações espaciais com a mesa (Quadro 12 linhas 8-3):

- a primeira relação espacial (**DirectRel_Per_Table**) é uma relação de direção e indica que a pessoa encontra-se à oeste da mesa (Quadro 12 linhas 0-2);
- a segunda relação espacial (**DistRel_Per_Table**) é uma relação de distância e indica que a distância é de 100 cm, e é definida como próxima ao objeto “**table**” (Quadro 12 linhas 3-7);

- a terceira relação espacial (***TopRel_Per_Table***) é uma relação topológica do tipo ***disjoint***, (não existe nenhuma intersecção nem limite entre dois objetos espaciais) (Quadro 12 linhas 18-20).

```

0  <geoR:DirectionRelation rdf:ID="DirectRel_Per_Table">
1    <geoR:isWestOf rdf:resource="#Table"/>
2  </geoR:DirectionRelation>

3  <geoR:DistanceRelation rdf:ID="DisRel_Per_Table">
4    <geoR:distance rdf:datatype="&xsd:string">100</geoR:distance>
5    <geoR:destination rdf:resource="#Table"/>
6    <geoR:hasQualitativeDistanceRelation rdf:resource="&geoR;close"/>
7  </geoR:DistanceRelation>

8  <spc:Person rdf:ID="Person_2">
9    <geoR:hasSpatialRelation rdf:resource="#DirectRel_Per_Table"/>
10   <geoR:hasSpatialRelation rdf:resource="#DistRel_Per_Table"/>
11   <geoR:hasSpatialRelation rdf:resource="#TopRel_Per_Table"/>
12   <geoD:shape rdf:resource="#Person_2Shape"/>
13 </spc:Person>

14 <spc:Object rdf:ID="Table">
15   <geoR:disjoint rdf:resource="#TopRel_Per_Table"/>
16   <geoD:shape rdf:resource="#TableShape"/>
17 </spc:Object>

18 <geoR:TopologicalRelation rdf:ID="TopRel_Per_Table">
19   <geoR:disjoint rdf:resource="#Table"/>
20 </geoR:TopologicalRelation>

```

Quadro 12 - Relações espaciais entre a pessoa e a mesa

4.4.3. Tempo

O “Tempo de Acesso a casa” (***Time_EnterHome***) é definido como parâmetro de entrada do serviço “Abrir Porta” (Quadro 3). Este parâmetro define o momento em que a pessoa é identificada e quer ingressar na casa. “Intervalo de tempo permitido” (***AllowedInterval***), e representa um intervalo de tempo permitido para a entrada na casa.

No Quadro 13, apresenta-se parte do código em OWL para definir duas entidades temporais, o instante de tempo “Tempo de Acesso a casa” (***Time_EnterHome***) e o intervalo de tempo “Intervalo de tempo permitido” (***AllowedInterval***).

```

0  <tme:Instant rdf:ID="Time_EnterHome">
1    <tme:inXSDDateTime rdf:datatype="&xsd;dateTime"
2      >2008-02-29T12:52:17</tme:inXSDDateTime>
3  </tme:Instant>

4  <tme:Interval rdf:ID="AllowedInterval">
5    <tme:hasEnd rdf:resource="#endAllowedInterval"/>
6    <tme:hasBeginning rdf:resource="#begAllowedInterval"/>
7  </tme:Interval>

8  <tme:Instant rdf:ID="begAllowedInterval">
9    <tme:inXSDDateTime rdf:datatype="&xsd;dateTime"
10     >2008-02-29T13:00:20</tme:inXSDDateTime>
11    <tme:inXSDDateTime rdf:datatype="&xsd;dateTime"
12     >2008-02-29T14:00:00</tme:inXSDDateTime>
13  </tme:Instant>

14 <tme:Instant rdf:ID="endAllowedInterval">
15   <tme:inXSDDateTime rdf:datatype="&xsd;dateTime"
16     >2008-02-29T16:00:00</tme:inXSDDateTime>
17 </tme:Instant>

```

Quadro 13 - Definição de entidades temporais

No Quadro 14 apresenta-se uma restrição em OWL para limitar o “tempo de Acesso a casa” de uma pessoa que esteja no intervalo de tempo “intervalo de tempo permitido”.

```

0  <owl:Class rdf:ID="AllowedInterval">
1    <rdfs:subClassOf rdf:resource="&time;#Interval" />
2  <rdfs:subClassOf>
3    <owl:Restriction>
4      <owl:onProperty rdf:resource="&time;#inside"/>
5      <owl:allValuesFrom rdf:resource="#Time_EnterHome"/>
6    </owl:Restriction>
7  </rdfs:subClassOf>
8  </owl:Class>

10 <owl:Class rdf:ID="Time_EnterHome">
11   <rdfs:subClassOf rdf:resource="&time;#Instant" />
12 </owl:Class>

```

Quadro 14 - Restrição do tempo de entrada á casa a um intervalo de tempo permitido

4.4.4. Atividades

O Quadro 15 apresenta a descrição da atividade “Rodrigo assiste televisão”. Para definição desta informação se utiliza a ontologia **HomeActivity**. Esta atividade acontece na sala (**LivingRoom_1**) (linha 6), estando definida em um intervalo de tempo (linha 5), sua prioridade é média, e tem como participante a Rodrigo (linhas 8).

Cada atividade tem associado um evento, neste caso utiliza-se o evento “Assistir Televisão” (linha 7) , cujo tipo é entretenimento (linha 12).

```

0 <act:Activity rdf:about="#Rodrigo_Watch_TV">
1   <act:activityName rdf:datatype="&xsd:string"
2     >Rodrigo_Watch_TV</act:activityName>
3   <act:isScheduled rdf:datatype="&xsd:boolean">false</act:isScheduled>
4   <act:priority rdf:resource="&act;medium"/>
5   <act:time rdf:resource="#Interval_Watch_TV"/>
6   <act:place rdf:resource="#LivingRoom_1"/>
7   <act:event rdf:resource="#Watch_TV"/>
8   <act:hasParticipant rdf:resource="&homePerson_data1;Person_5"/>
9 </act:Activity>

10 <act:Event rdf:about="#Watch_TV">
11   <act:eventName rdf:datatype="&xsd:string">Watch TV</act:eventName>
12   <act:hasType rdf:resource="&act;Entertainment"/>
13 </act:Event>

```

Quadro 15 - Descrição da atividade “Rodrigo Assiste Televisão”

4.4.5. Dispositivos

A descrição dos dispositivos consiste em estabelecer o tipo e as características. Na Tabela 3 é apresentada a descrição de alguns dos dispositivos para o Estudo de Caso. Para a descrição dos dispositivos são utilizadas as ontologias **Device**, **DeviceProfile** e a ontologia específica de domínio **HomeDevice**.

Tabela 3 - Descrição de dispositivos para o Estudo de Caso

Tipo	Nome	Descrição
Computacionais	Computador	Um computador pessoal
	Celular	Um dispositivo móvel
Sensor	Sistema de Identificação	Reconhece uma pessoa
	Sensor de Localização	Fornecer a localização de uma pessoa ou objeto na casa
	Sensor de Iluminação	Mede a iluminação do ambiente
Atuador	Abertura de portas	Permite automaticamente abrir as portas da casa
	Lâmpadas	Controlam a iluminação do ambiente
Eletrônicos	Intercomunicador	Estabelece a comunicação entre pessoas
	Televisão	Eletrodoméstico de entretenimento

4.4.6. Privacidade

Segundo o Estudo de Caso (Seção 4.2), é definida uma política de privacidade, a qual estabelece que só as pessoas autorizadas possam ingressar na casa. No Quadro 16 apresenta-se a definição da política de privacidade para controlar o ingresso na casa “**Enter_Home**” (linha 5).

```

0  <pol:Police rdf:about="#Police_Home_Entrance">
1    <pol:creationDate rdf:datatype="&xsd;dateTime"
2      >2008-03-27T22:05:24</pol:creationDate>
3    <pol:validade rdf:datatype="&xsd;dateTime"
4      >2008-04-27T22:05:27</pol:validade>
5    <pol:controls rdf:resource="&homeAct_1;Enter_Home"/>
6    <pol:creator rdf:resource="&homePerson_data1;Person_1"/>
7  </pol:Police>

```

Quadro 16 - Política de privacidade para ingressar à casa

Para definição das pessoas autorizadas da política de privacidade definida anteriormente, utiliza-se a regra SWRL definida no Quadro 17 (a regra está expressa na sintaxe XML do SWRL) (HORROCKS et al., 2004).

```

0  <ruleml:imp>
1    <ruleml:rlabel ruleml:href="#PolicesWRL_Home_Entrance" />
2    <ruleml:body>
3      <swrlx:classAtom>
4        <owlx:Class owlx:name="&pol;Police" />
5        <ruleml:var>#pol</ruleml:var>
6      </swrlx:classAtom>
7      <swrlx:individualPropertyAtom swrlx:property="&pol;controls">
8        <ruleml:var>pol</ruleml:var>
9        <ruleml:var>"&homAct_1;Enter_Home"</ruleml:var>
10     </swrlx:individualPropertyAtom>
11     <swrlx:classAtom>
12       <owlx:Class owlx:name="&homePer;Resident" />
13       <ruleml:var>#per</ruleml:var>
14     </swrlx:classAtom>
15   </ruleml:body>
16   <ruleml:head>
17     <swrlx:individualPropertyAtom
18       swrlx:property="&pol;authorizedPersons">
19       <ruleml:var>#pol</ruleml:var>
20       <ruleml:var>#per</ruleml:var>
21     </swrlx:individualPropertyAtom>
22   </ruleml:head>
23 </ruleml:imp>

```

Quadro 17 - Regra SWRL para controlar o ingresso na casa somente a residentes

4.4.7. Preferência

No Estudo de Caso foi apresentado que o ambiente obtém as preferências de Alex assumindo que é a sua primeira vez na casa, portanto não existe perfil dele. No dispositivo pessoal dele, encontram-se seu perfil de preferências, o qual é repassado ao sistema. O perfil de preferências inclui desde canais de TV, comida, até o tipo de música. O sistema captura tais preferências e as transforma em ontologias.

No Quadro 18 é apresentada a descrição da preferência da Rodrigo por assistir televisão.

```

0  <pref:EventPreference rdf:about="#Rodrigo_Watch_TV_Preference">
1    <pref:preferenceName rdf:datatype="&xsd:string"
2      >Rodrigo_Watch_TV_Preference</pref:preferenceName>
3    <pref:preferEvent rdf:resource="&homeAct_1;Watch_TV"/>
4    <pref:preferenceOf rdf:resource="&homePerson_data1;Person_5"/>
5    <pref:qualitativeDescription rdf:resource="&pref;medium"/>
6  </pref:EventPreference>

```

Quadro 18 - Preferência de Rodrigo por assistir televisão

A descrição da preferência de Rodrigo definida anteriormente não define em que situação Rodrigo prefere assistir televisão. Para definir este tipo de situações são utilizadas regras SWRL. Por exemplo, consideremos o enunciado seguinte: “*Rodrigo prefere assistir programas de TV no seu tempo livre nas tardes*”.

No Quadro 19 é apresentada a preferência de Rodrigo por assistir televisão, no formato que utiliza o *plugin* SWRLTab (SWRLTAB ..., 2007) do Protege (THE PROTÉGÉ ..., 2008) para edição das regras SWRL.

```

0  pref:Preference(?pref)  ^
1  act:Activity(?act)  ^
2  pref:Event(?eve)  ^
3  act:event(?act, ?eve)  ^
4  act:hasType(?eve, FREE_TIME)  ^
5  act:time(?act, ?tme)  ^
6  tme:inside(?tme, AFTERNOON)  ^
7  per:Person(?per)  ^
8  act:hasParticipant(?act, ?per)  ^
9  pref:hasPreference(?per, ?pref)  →
10 pref:preferEvent(?pref, homeAct_1:Watch_TV)

```

Quadro 19 - Preferência de Rodrigo por assistir televisão nas tardes e em seus tempos livres

5. RESULTADOS

Neste capítulo são descritos os materiais utilizados, os testes realizados e a comparação do modelo de contexto proposto com outros trabalhos.

5.1. Materiais

Nesta seção, são apresentadas as linguagens e ferramentas utilizadas neste trabalho.

5.1.1. Linguagens

Neste trabalho são utilizadas as seguintes linguagens:

- é adotada a linguagem OWL (SMITH; WELTY; MCGUINNESS, 2004) no dialeto DL para a modelagem de contexto;
- e a ontologia OWL-S (MARTIN et al., 2004) para a modelagem de serviços.;
- para a modelagem das preferências e as políticas de privacidade são definidas regras na linguagem SWRL (HORROCKS et al., 2004).

5.1.2. Ferramentas

Para a criação das ontologias utiliza-se o Protégé (THE PROTÉGÉ ..., 2008) o qual permite a utilização de *plugins*. As versões do Protégé e os *plugins* utilizados neste trabalho são os seguintes:

- o Protege 3.4 com o *plugin* GrOWL (GROWL ..., 2008) permite a visualização gráfica das ontologias e com o *plugin* SWRLTab (SWRLTAB ..., 2007), permite a edição visual de regras SWRL;
- o Protege 3.3.1 com o *plugin* OWL-S Editor (THE OWL-S ..., 2004), permite a descrição semântica dos serviços em OWL-S em um entorno visual;
- o Protege 4.0 permite a construção de ontologias OWL-DL com suporte total para o OWL-DL e o OWL 1.1 (PATEL-SCHNEIDER; HORROCKS, 2006).

Para obtenção dos resultados foram utilizados o Protégé e a máquina de inferência Pellet (PELLET ..., 2008), também foi utilizado o Eclipse 3.2 e as seguintes bibliotecas Java:

- o OWL-API (OWL API ..., 2008) é uma livreria para manipulação de ontologias em OWL-DL e o OWL 1.1. Esta biblioteca também fornece suporte para o SWRL e unida juntamente com o Pellet, possibilita inferência semântica em ontologias e regras em SWRL;
- a máquina de inferência Pellet fornece suporte para inferência semântica no OWL-DL.

As tarefas de inferência do Pellet são os seguintes:

- verificação de consistência, ou seja, determina que a ontologia não possua dados contraditórios;
- classificação, a qual consiste na criação de uma completa hierarquia de classes;
- realização, que se refere ao processo através do qual se encontra a classe mais específica à qual uma instância pertenceria;
- suporte limitado a regras codificadas em SWRL.

5.2. Testes

Nesta seção serão apresentados os testes para avaliar a utilização da inferência semântica no modelo de contexto proposto.

Os testes 1 e 2 são utilizados para avaliar a inferência semântica. No primeiro teste são deduzidos novos dados e no segundo teste é verificada a consistência da informação. O teste 3 mostra a interoperabilidade semântica, na qual é realizado um mapeamento entre duas classes similares definidas em ontologias diferentes e também é definida a equivalência entre duas instâncias.

5.2.1. Critérios

Os critérios considerados utilizados na comparação são os seguintes:

5.2.1.1. *Inferência semântica*

Em um Ambiente Pervasivo, a informação de contexto pode estar incompleta ou ser inconsistente, devido ao sensoriamento impreciso (CHEN; FININ; JOSHI, 2004). Uma das vantagens da modelagem de contexto utilizando ontologias é a possibilidade de inferir nova informação (CHEN; FININ; JOSHI, 2004) (GU et al., 2004)(BULCÃO; PIMENTEL, 2005).

5.2.1.2. Interoperabilidade

Uma das características do Ambiente Pervasivo é o acesso ubíquo (Seção 2.1.2) uma vez que existe uma variedade de dispositivos, serviços e aplicações, os quais devem interoperar entre si. A abordagem deste trabalho é a descrição semântica do contexto de modo a garantir um conhecimento compartilhado. A linguagem OWL permite realizar mapeamentos entre termos distintos das ontologias, de forma que possam ser integradas (OWL ..., 2004).

5.2.2. Teste 1

Neste teste será analisada a inferência automática de novas informações.

5.2.2.1. Dados de teste

Na Tabela 4

Tabela 4 apresentam-se dados incompletos da modelagem das pessoas, sua idade e sexo, assim como seus relacionamentos. Tais dados correspondem à modelagem de pessoas do Estudo de Caso (Seção 4.4.1).

Para a definição dos relacionamentos entre as pessoas a ontologia **HomePerson** (Seção 3.3.2) define as seguintes propriedades tipo de objeto: “pai de” (**parentOf**), “filho de ” (**childOf**), “esposo(a) de” (**spouseOf**), “irmão(ã) de” (**siblingOf**), “colega de” (**colleageOf**), “amigo(a) de” (**friendOf**). Tais propriedades são sub-propriedades da propriedade “conhece” (**knows**).

Tabela 4 - Instâncias exemplo da modelagem de pessoas

Nome	Idade	Sexo	Relações
Ricardo	40	M	Pai de Alex
Mariela	38	F	Mãe de Andrea Esposa de Ricardo
Alex	16	M	Filho de Mariela
Andréa	14	F	Filha de Ricardo Irma de Alex
Rodrigo	36	M	Colega do trabalho de Ricardo
Carlos	16	M	Amigo de Alex

5.2.2.2. Resultados

Utilizando a máquina de inferência Pellet na informação apresentada na Tabela 4 obtemos nova informação. Na Tabela 5 são apresentadas as propriedades tipo de objeto e as classes relacionadas a Ricardo, depois de se aplicar a inferência.

Tabela 5 - Propriedades tipo objeto e classes de Ricardo

Propriedades tipo de objeto de Ricardo
<p>Esposo de Mariela</p> <p>Pai de Alex e Andrea</p> <p>Colega de Rodrigo</p> <p>Conhece a Mariela, Andrea, Alex e Rodrigo</p>
Classes
<p>Homem</p> <p>Adulto</p> <p>Pai</p>

Na Figura 17 é apresentada a obtenção dos resultados do teste 1; utilizando-se o Protege 4.0, no qual são detalhadas as classes, as propriedades tipo de objeto, a informação definida e a informação deduzida.

Figura 17 - Janela dos dados inferidos do Teste - 1 utilizando o Pellet 4.0

5.2.2.3. Explicação dos resultados

Explicam-se os resultados sobre Ricardo, descritos na Tabela 5 da seguinte forma:

- foi definido que Mariela é “esposa de” Ricardo, já que “esposa de” é uma propriedade simétrica. Portanto, é deduzido automaticamente que Ricardo é “esposo de” Mariela;

- foi definido que Ricardo é “pai de” Alex, e que Andrea é “filha de” Ricardo já que “pai de” é propriedade inversa com “filho(a) de”. Portanto, é deduzido automaticamente que Ricardo é também “pai de” Andrea;
- foi definido que Rodrigo é “colega de” Ricardo já que “colega de” é uma propriedade simétrica. Portanto, é deduzido automaticamente que Ricardo é “colega de” Rodrigo;
- já que todas as propriedades de relacionamentos são sub-propriedades da propriedade “conhece” (**knows**). Portanto, é deduzido que Ricardo “conhece” a Mariela, Andrea, Alex e Rodrigo;
- foi definido que o sexo de Ricardo é Masculino (**MaleSex**). Descreve-se que uma pessoa é “Homem” se seu sexo é “Masculino” (**MaleSex**) e “Mulher” se seu sexo é “Feminino”. Portanto, é deduzido que Ricardo é “Homem”;
- foi definido que a idade de Ricardo é 40 anos. Descreve-se uma pessoa como Adulta “**AdultPerson**” se sua idade é maior de 18 anos. Portanto, é deduzido que Ricardo é uma pessoa adulta;
- foi definido que Ricardo é “pai de” Alex, é inferido que também é “pai de” Andrea. Descreve-se que uma pessoa é “Pai” (**Parent**) se tem pelo menos um filho. Portanto, é deduzido que Ricardo é “Pai”.

5.2.3. Teste 2

Neste teste será analisada como a utilização de semântica garante um conhecimento consistente.

5.2.3.1. *Dados de teste*

Uma pequena modificação nos dados do teste 1 foi introduzida, no qual é declarado intencionalmente que Ricardo é “Mulher”.

5.2.3.2. *Resultados*

O resultado da verificação da consistência indicou inconsistência nos dados.

5.2.3.3. *Explicação dos resultados*

A definição do gênero de Ricardo como masculino e a declaração dele como mulher é inconsistente, uma vez que a definição de uma pessoa de gênero masculino implica em que essa seja homem segundo a ontologia.

5.2.4. Teste 3

A ontologia **Space** permite definir a localização de pessoas e objetos em um determinado espaço. Por razões de modularidade a ontologia **Space** não importa a ontologia **Person**. Na Figura 18, apresentam-se a hierarquia de classes da ontologia **HomePerson** (izquierda) e a ontologia **Space** (direita).

A ontologia **HomeSpace** importa tanto a ontologia **Space** quanto a ontologia **HomePerson** e realiza um mapeamento da classe **Person** definida na ontologia **Space** (Quadro 20 linha 2) e na ontologia **Person** (Quadro 20 linha 0).

Este mapeamento é definido mediante uma declaração de equivalência (Quadro 20 linha 3) entre as duas classes, de tal forma, que nós podemos interoperar entre as duas ontologias e ter um conhecimento compartilhado, no qual é possível ter informações sobre espaços e pessoas.

Figura 18 - Diagrama de atividades do serviço "Ingressar na casa"

```

0 <owl:Class rdf:about="&per;Person">
1   <owl:equivalentClass>
2     <owl:Class rdf:about="&spc;Person" />
3   </owl:equivalentClass>
4 </owl:Class>

```

Quadro 20 - Definição de equivalência entre duas propriedades de ontologias distintas

5.2.4.1. Dados de teste

Uma vez definido o mapeamento descrito anteriormente é possível a definição da informação de propriedades de Rodrigo e sua respectiva localização. No Quadro 21 é descrito que a cozinha contém Rodrigo.

No teste 1 foram definidos relacionamentos de pessoas (Tabela 4

Tabela 4), das quais uma delas é o Rodrigo. Entretanto no (Quadro 21) é definido informação sobre a localização de Rodrigo.


```

0  <spc:Room rdf:ID="kitchen">
1    <spc:name rdf:datatype="&xsd:string">Kitchen</spc:name>
2    <spc:contains rdf:resource="#Spc_Per_1"/>
3  </spc:Room>

4  <spc:Person rdf:ID="Spc_Per_1">
5    <spc:name rdf:datatype="&xsd:string">Rodrigo</spc:name>
6  </spc:Person>

```

Quadro 21 - Instâncias exemplo da localização de uma pessoa e um objeto

No Quadro 22, acrescentam-se as seguintes descrições sobre Rodrigo e sua localização:

- foi definida uma equivalência entre o Rodrigo definido na Tabela 4 (**Spc_Per_1**) utilizando a ontologia **Space** e o Rodrigo do Quadro 21 (**Person_5**) utilizando a ontologia **Person** (linhas 0-2);
- foi definido como país Brasil, o qual contém a cidade de São Paulo (linhas 3-5, 14-16);
- O andar "**Floor_1**" contém a cozinha (linhas 6-9).
- foi definida também uma casa "**Home_Test**" localizada em São Paulo e que contém o andar "**Floor_1**" (linhas 10-13);

```

0  <rdf:Description rdf:about="&homPer_1;Person_5">
1    <owl:sameAs rdf:resource="&spc_1;Spc_Per_1"/>
2  </rdf:Description>
3  <space:Country rdf:about="#Brazil">
4    <space:name rdf:datatype="&xsd:string">Brazil</space:name>
5  </space:Country>
6  <space:Floor rdf:about="#Floor_1">
7    <space:name rdf:datatype="&xsd:string">Floor 1</space:name>
8    <space:contains rdf:resource="&spc_1;kitchen"/>
9  </space:Floor>
10 <rdf:Description rdf:about="#Home_Test">
11   <space:contains rdf:resource="#Floor_1"/>
12   <space:locatedIn rdf:resource="#SaoPaulo"/>
13 </rdf:Description>
14 <space:City rdf:about="#SaoPaulo">
15   <space:name rdf:datatype="&xsd:string">Sao Paulo</space:name>
16 </space:City>

```

Quadro 22 - Instâncias exemplo dos dados de uma pessoa e sua localização

5.2.4.2. *Resultados*

Entre os resultados obtém-se que Rodrigo tem 36 anos, é homem, colega de trabalho de Ricardo e está localizado em São Paulo – Brasil.

5.2.4.3. *Explicação dos resultados*

É obtido esse resultado devido ao mapeamento realizado entre distintas ontologias (*Space* e *Person*) com termos similares. De forma que foi possível ter um conhecimento compartilhado.

5.3. Comparação com outras abordagens

Nesta seção apresenta-se a comparação do modelo de contexto proposto com outras abordagens. Primeiramente são descritos os critérios adotados, assim como uma descrição de cada uma das abordagens e a tabela de comparação de cada uma das abordagens considerando os critérios de comparação descritos. Finalmente é dada uma análise da comparação da abordagem apresentada neste trabalho com outras abordagens.

5.3.1. Critérios de comparação

Os critérios adotados para a comparação são os seguintes:

- **Modularidade:** Um modelo de contexto deve estar estruturado em módulos de maneira que possa ser estendido de uma maneira prática;
- **Interoperabilidade:** Uma aplicação dos Ambientes Pervasivos deve interagir com dispositivos ubíquos heterogêneos. Um dos principais motivos de modelar contexto utilizando ontologias é a necessidade de interoperabilidade entre sistemas e dispositivos;
- **Elementos contextuais:** Os quais são os elementos básicos considerados na modelagem de contexto. Estes elementos são a pessoa, o espaço, o tempo, as atividades, os dispositivos entre outros;
- **Privacidade:** A informação de contexto inclui toda informação útil para dar uma resposta a uma pessoa. É necessário fornecer políticas de privacidade para acesso aos dispositivos e à informação;
- **Preferencial:** A resposta do Ambiente Pervasivo depende do perfil da pessoa, suas preferências e necessidades. Um serviço personalizado garante uma maior satisfação pessoal;
- **Dinâmica:** Um Ambiente Pervasivo é altamente dinâmico, tanto pela mobilidade dos dispositivos quanto pela aparição/desaparição dos dispositivos e as preferências das pessoas. O Ambiente Pervasivo tem que considerar esse aspecto para atuar de modo autônomo e se auto-configurar.

5.3.2. Abordagens para a modelagem de contexto e serviços

Neste item, apresentamos sete abordagens, identificadas como A1, A2, ..., A7, as quais serão comparadas com a abordagem A8, proposta neste trabalho.

- **A1:** Uma abordagem interessante é a linguagem gráfica *Context Modelling Language* (CML) (HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002). O CML permite a modelagem da qualidade, dependências, persistência e fonte de contexto. Henricksen e Indulska (2006) apresentam um modelo de

preferência, utilizado com o CML, que assinala um *ranking* das preferências, as quais podem ser agrupadas em conjuntos e combinadas com políticas;

- **A2:** Gu et al. (2004), descreve um modelo semântico de contexto utilizando ontologia. Este modelo de contexto tem ontologias de nível superior e ontologias para um domínio em particular. As ontologias de nível superior incluem a definição de pessoas, localização, atividade e entidade computacional. Os autores utilizam propriedades na linguagem OWL para definir parâmetros de qualidade, classificação, dependência e qualidade de contexto;
- **A3:** Chen, Finin e Joshi (2005) apresentam o Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA), um modelo semântico de contexto que está formado por ontologias dependentes e independentes de domínio. O SOUPA inclui vocabulários para representar e raciocinar sobre eventos temporais. Utiliza a linguagem de políticas de privacidade Rei (REI ..., 2005) para definir: direitos, proibições e obrigações. O SOUPA inclui também vocabulários para descrever entidades computacionais e as pessoas como agentes, suas metas, planos, desejos e intenções;
- **A4:** Bulcão e Pimentel (2005) descrevem um modelo de contexto que considera as dimensões contextuais e permite a modelagem de pessoas, espaço, tempo, dispositivos e atividades. Posteriormente seu modelo semântico de contexto SeCoM Semantic Context Model (Bulcão, 2006) é estruturado em módulos e possui ontologias independentes e dependentes de domínio;
- **A5:** Christopoulou, Goumopoulos e Kameas (2005), apresentam a ontologia *GAS Ontology*, a qual fornece uma linguagem comum para a comunicação e colaboração entre dispositivos heterogêneos que constituem um Ambiente Pervasivo. Esta abordagem considera a autonomia dos dispositivos mediante o modelo “*Plug Synapse*” o qual permite a composição de serviços e define a colaboração entre os dispositivos, além de permitir criar e eliminar associações entre eles;

- **A6:** Broens et al. (2004) utilizam SWS; sua abordagem é a descrição de um algoritmo de descoberta de serviços semânticos relacionados a Ambientes Pervasivos;
- **A7:** Mokhtar et al. (2006) modelam serviços e tarefas do usuário como SWS utilizando a ontologia de serviços OWL-S, a qual é estendida com informação contextual. Os autores apresentam um algoritmo de descoberta semântica de serviços em Ambientes Pervasivos;
- **A8:** O modelo de contexto proposto (PONCE et al., 2006) apresenta uma divisão modular e em dois níveis; no primeiro nível, encontram-se as ontologias independentes de contexto e no segundo as dependentes de domínio. As ontologias incluem a definição de pessoas, espaço, tempo, atividades e políticas de privacidade.

Posteriormente, o modelo de contexto proposto inclui a modelagem semântica de serviços cientes ao contexto semânticamente utilizando o OWL-S (PONCE et al., 2007). O modelo de contexto apresentado neste trabalho, denominado OMC, inclui também a modelagem de políticas de privacidade e preferências utilizando a linguagem SWRL.

5.3.3. Tabelas de comparação

As comparações entre as diferentes abordagens são apresentados na Tabela 6. As abordagens são qualificadas da seguinte forma:

- “✓” - quando o critério de comparação é definido de uma forma adequada;
- Em branco - quando não é considerado o critério de comparação.

Tabela 6 - Diferentes abordagens para modelagem de contexto

	A1	A2	A3	A4	A5	A6	A7	A8
Modularidade	✓	✓	✓	✓	✓	✓	✓	✓
Interoperabilidade		✓	✓	✓	✓	✓	✓	✓
Qualidade	✓	✓						
Elementos contextuais	✓	✓	✓	✓				✓
Privacidade	✓		✓					✓
Preferencial	✓							✓
Dinâmica					✓	✓	✓	✓

5.3.4. Análise da comparação

Os resultados da análise da comparação são os seguintes:

- a abordagem A1 apresenta o CML. Esta é uma das abordagens mais completas, considerando os critérios de comparação a principal desvantagem desta proposta é que não utiliza ontologias, o que dificulta a interoperabilidade e inferência de contexto. A modelagem de preferências definido por Henriksen e Indulska (2006), serve como base para a construção da ontologia **Preference**.
- as abordagens A2, A3 e A4 utilizam semântica como técnica para a modelagem de contexto, seu modelo de contexto é modular e inclui ontologias dependentes e independentes de domínio, mas não consideram as preferências das pessoas e o dinamismo do ambiente. A abordagem A4 utiliza a linguagem Rei para definir políticas de privacidade, esta é uma linguagem robusta, mas não é definida em OWL, o que

dificulta a obtenção de resultados. Esta linguagem serve como base para a definição da ontologia ***Police***.

- a abordagem A5 utiliza ontologias para a modelagem de contexto; esta abordagem não considera a modelagem de tempo, espaço, privacidade e preferências, por tanto, não é uma ontologia adequada. Um ponto forte desta abordagem é o suporte para composição de serviços em um Ambiente Dinâmico, mediante o uso da GAS *ontology*. Com tudo, esta ontologia não é referenciada em outros trabalhos. A GAS *ontology* serve como inspiração na integração da modelagem de contexto e a modelagem de serviços utilizando OWL-S. Esta integração foi definida em (PONCE et al, 2007) e segundo o nosso conhecimento essa integração não foi apresentada anteriormente;
- as abordagens A6 e A7, não têm um modelo para a modelagem de contexto comparado com outras abordagens, mas nestas abordagens são modelados os serviços em um Ambiente Pervasivo como SWS utilizando a linguagem OWL-S, o que fornece suporte ao dinamismo de ambiente, permitindo a composição e descoberta automática de serviços.

A abordagem apresentada neste trabalho (A8) OCM, considera a maior parte dos critérios de comparação analisados como segue:

- apresenta um conjunto de ontologias em dois níveis (Seção 3.2) é estruturada em módulos. Para a modelagem de uma determinada informação, requer-se somente utilizar uma parte das ontologias do OMC, assim como foi apresentado na modelagem de pessoas, espaço, atividades entre outros (Seção 4.4);
- permite a modelagem dos elementos contextuais (Seções 3.3-3.7) e apresenta uma melhoria com respeito as outras abordagens já que integra características das outras abordagens. Nesse trabalho as ontologias *Space* e

Device são mais robustas que as ontologias *Space* e *Device* das ontologias A2, A3 e A4;

- permite a modelagem de políticas de privacidade (Seção 3.8);
- permite a modelagem de preferências (Seção 3.9);
- descreve a integração da modelagem de contexto e a integração desta com a modelagem de serviços, dando suporte ao dinamismo do Ambiente Pervasivo (Seção 3.10);
- permite interoperabilidade como foi apresentado no Teste 3 (Seção 5.2.4).

6. CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo serão descritas as conclusões do trabalho, os trabalhos futuros e as publicações realizadas.

6.1. Conclusões

A seguir são descritas as conclusões:

- verificou-se que OMC pode ser aplicado facilmente, tal como foi realizado no o Estudo de Caso do domínio de Uma Casa Inteligente (Capítulo 4);
- este modelo apresenta melhora na modelagem de contexto em comparação com outros trabalhos, pois permite a modelagem de políticas de privacidade e preferências das pessoas em SWRL, e a modelagem de serviços utilizando OWL-S (Seção 5.3);
- verificou-se que a utilização de ontologias para modelagem de contexto fornece uma modelagem de contexto adequada, em função da inferência semântica que pode ser aplicada e a interoperabilidade, conforme demonstram os testes realizados (Seção 5.2);
- verificou-se de fato que o modelo de contexto proposto OMC é um modelo de contexto adequado para Ambientes Pervasivos;

6.2. Limitações

A seguir serão descritas as limitações deste trabalho:

- não utiliza informação de contexto fornecida por dispositivos reais. Para a implementação real de um Ambiente Pervasivo será necessário a utilização

de dispositivos inteligentes e a criação das interfaces para interação com tais dispositivos;

- foi considerado principalmente a característica ciente de contexto de um Ambiente Pervasivo. A implementação real de um Ambiente Pervasivo requer a utilização conjunta das outras três características: comportamento “inteligente”, acesso ubíquo e interfaces naturais;

6.3. Trabalhos futuros

Destacam-se como trabalhos futuros os seguintes:

- aplicação do OMC em domínios diferentes como Conferências Inteligentes, Automóveis Inteligentes, Comércio Pervasivo, entre outros. Acredita-se que sua estrutura modular e a divisão em dois níveis facilitarão esta aplicação em outros domínios;
- implementação de um algoritmo *matching*, para realizar descoberta semântica, de maneira que se forneça suporte ao dinamismo do Ambiente Pervasivo.

6.4. Publicações

6.4.1. Publicação em 2006

PONCE, E.; BEINGOLEA, J.; ORRILLO, H. KOFUJI, S. Manipulación de Información de Contexto de una Casa Inteligente usando ontologias. In: V Congreso de Informática y Sistemas de Sudamérica COISIS 2006, 5. Peru, 2006, Anais..., Tacna, 2006, 2006, p. 1-8.

Neste trabalho é construído um modelo semântico de contexto que permite a modelagem de pessoas, espaço, tempo, atividade, dispositivos e políticas de privacidade.

6.4.2. Publicação em 2007

PONCE, E.; CASSIO V. PRAZERES, C. AND KOFUJI, S.; TEIXEIRA, C.; PIMENTEL, M.. SeCoAS: An Approach to Develop Semantic and Context-Aware Available Services, In: XIII Brazilian Symposium on Multimedia and the Web , 13. Rio Grande do Sul, Brazil, 2007, Anais..., Gramado, 2007, p. 182-189.

Neste trabalho é apresentada a integração da modelagem de serviços utilizando o OWL-S e a modelagem de contexto usando OWL. Esta integração apresenta o uso de parâmetros contextuais e a definição do comportamento do Ambiente Pervasivo.

REFERÊNCIAS

ABOWD, G.; MYNATT D. Charting past, present, and future research in ubiquitous computing. **ACM Trans. Comput.-Hum. Interact (ACM Press)**, v. 7, p. 29-58, 2000.

ALSOS, O.; SVANAES, D. Interaction techniques for using handhelds and PCs together in a clinical setting. In: NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction, 6. 2006, Oslo, Norway. **Anais...** NY, USA: ACM. 2006. p. 125-134.

AMIGO Ambient intelligence for the networked home environment Homepage. 2008. Disponível em: <<http://www.hitech-projects.com/euprojects/amigo/>>. Acesso em: 08 Maio 2008.

ANTONIOU, A.; HARMELEN. V. A Semantic Web Premier. MIT Press, 2004.

BARKHUUS, L.; DEY, A.. Is context-aware computing taking control away from the user? Three levels of interactivity examined. **Lecture Notes in Computer Science (Springer-Verlag)**, v. 2864, p. 149-156, 2003.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O., The Semantic Web, **Scientific American**, May 2001.

BRAY, T.; PAOLI, J.; SPERBERG-C, C. Extensible Markup Language (XML) 1.1 (Second Edition) Homepage. 2006. Disponível em: <<http://www.w3.org/TR/xml11/>>. Acesso em: 08 Maio 2008.

BRICKLEY, D. Basic Geo (WGS84 lat/long) Vocabulary Homepage. 2003. Disponível em: <<http://www.w3.org/2003/01/geo/>>. Acesso em: 08 Maio 2008.

BRICKLEY, D.; MILLER, L. FOAF Vocabulary Specification 0.9 Homepage. 2007. Disponível em: <<http://xmlns.com/foaf/0.1/>>. Acesso em: 08 Maio 2008.

BROENS, T; POKRAEV S.; SINDEREN M.; KOOLWAAIJ J.; DOCKHORN P. Context-Aware, Ontology-Based Service Discovery. In. Second European Symposium, EUSAI 2004, 2. 2004, **Anais...** Lecture Notes in Computer Science (Springer Verlag), 2004. p. 72-83.

BROLL, G.; SIORPAES S.; RUKZIO E.; PAOLUCCI M.; HAMARD J.; WAGNER M.; SCHMIDT A. Supporting Mobile Service Usage through Physical Mobile Interaction.

In. PERCOM '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications, 7. 2007. **Anais...**, Washington, DC, USA: IEEE Computer Society, 2007. p. 262-271.

BULCÃO R. **Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto**. 2006. 219 p. Tese (Doutorado) - Instituto de Ciências Matemáticas e de Computação (ICMC) – Universidade de São Paulo São Carlos. São Paulo, 2006.

BULCÃO R.; PIMENTEL, M. Toward a Domain-Independent Semantic Model for Context-Aware Computing. In. LA-WEB '05: Proceedings of the Third Latin American Web Congress, 5. 2005, **Anais...** Washington, DC, USA.: IEEE Computer Society, 2005. p. 61.

CAPRA, L.; EMMERICH, W.; MASCOLO, C. Reflective Middleware Solutions for Context-Aware Applications. **REFLECTION '01: Proceedings of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns (Springer-Verlag)**, v. 2192, p. 126-133, London, UK, 2001.

CHEN, G.; KOTZ, D. **A Survey of Context-Aware Mobile Computing Research**, Hanover, NH, USA: Dartmouth College, 2000, paginação irregular. (Science Technical Report TR2000-381).

CHEN, H.; FININ, T.; JOSHI, A. Semantic Web in the Context Broker Architecture. In: PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04), 4. 2004, **Anais ...** Washington, DC, USA: IEEE Computer Society, 2004. p. 277.

CHEN, H.; PERICH F.; FININ T.; JOSHI A. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), 1. 2004, **Anais...** Los Alamitos, CA, USA: IEEE Computer Society, 2004. p.258-267.

CHRISTENSEN, E.; CHRISTENSEN, E.; CURBERA F.; MEREDITH, G.; WEERAWARANA, S. Web Services Description Language (WSDL) 1.1. Homepage. 2001. Disponível em: <<http://www.w3.org/TR/wsdl>>. Acesso em: 08 Maio 2008.

CHRISTOPOULOU, E.; GOUMOPOULOS, C.; KAMEAS A., An ontology-based context management and reasoning, In: Process for UbiComp applications. Proceedings of the 2005 Joint conference on Smart objects and ambient intelligence, 2005, **Anais...** Grenoble, France, , 2005, p. 265 – 270.

CORRADINI, A. WESSON, R.; COHEN E. A Map-Based System Using Speech and 3D Gestures for Pervasive Computing. In: ICMI '02: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, 2. 2002, **Anais...** Los Alamitos, CA, USA: IEEE Computer Society, 2004, p. 191.

D3.3: Amigo Middleware Core Enhanced: Prototype Implementation & Documentation (Technical report, Amigo Project Deliverable), 2006.

DAML-S. DAML-S Research Project. Homepage Disponível em :<<http://www.ksl.stanford.edu/projects/DAML/damls.shtml>>. Acesso em: 08 Maio 2008.

DAVIS, I.; VITIELLO, E. RELATIONSHIP: A vocabulary for describing relationships between people Homepage. 2005. Disponível em: <<http://vocab.org/relationship/>>. Acesso em: 08 Maio 2008.

DEY, A.; SALBER, D.; ABOWD, G. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Human-Computer Interaction Journal**, v. 16(2), p. 97-166, [S.I.], 2001.

EDWARDS, W. K. Discovery Systems in Ubiquitous Computing. **IEEE Pervasive Computing**, v. 5, p. 70-77, Piscataway, NJ, USA, 2006.

FENG, L.; APERS P.; JONKER, W. Towards Context-Aware Data Management for Ambient Intelligence. In: Proceedings of the 15th International Conference on Database and Expert Systems Applications (DEXA 2004), 15. 2004, Zaragoza – Spain. **Anais...** Berlin:Springer-Verlag, 2004. p. 422-431.

GEOONTOLOGIES Homepage. 2004. Disponível em: <<http://www.mindswap.org/2004/geo/geoOntologies.shtml>>. Acesso em: 08 Maio 2008.

GRIMM, R.; DAVIS J.; LEMAR, E.; MACBETH, A.; SWANSON, S.; ANDERSON, T.; BERSHAD, B; BORRIELLO, G.; GRIBBLE, S.; WETHERALL, D. System support for pervasive applications. **ACM Trans. Comput. Syst.** v. 22, p. 421-486, NY, USA, 2004.

GROWL Homepage. 2008. Disponível em: <<http://www.uvm.edu/~skrivov/growl/>>. Acesso em: 08 Maio 2008

GRUBER, T. A translation approach to portable ontology specifications. **Knowl. Acquis.**, v. 5, p. 199-220, London, UK, 1993.

GRÜNINGER, M. Designing and Evaluating Generic Ontologies. In: ECAI96's Workshop on Ontological Engineering, 1996, **Anais...** Budapest, 1996, p. 53-64.

GRÜNINGER, M.; FOX, M.. Methodology for the design and evaluation of ontologies. In: Skuce D, editor. IJCAI96 Workshop on basic ontological issues in knowledge sharing, 1995, **Anais...** Montreal: AAAI Press, p. 6.1-10, 1995.

GU, T.; WANG, W.; PUNG, H.; ZHANG, D. An Ontology-based Context Model in Intelligent Environments. In: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, 2004, **Anais...** San Diego, California, USA, p. 270-275, 2004.

HARIHAR, K.; KURKOVSKY, S.. Using Jini to enable pervasive computing environments. In: ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference, 2005, Kennesaw, Georgia, **Anais...** New York, NY, USA: ACM, 2005, p. 188-193.

HELAL, S. The Gator Tech Smart House: A Programmable Pervasive Space. Computer. **IEEE Computer Society Press**, v. 38, p. 50-60, Los Alamitos, CA, USA, 2005.

HENRICKSEN, K.; INDULSKA, J.; RAKOTONIRAINY A. Modeling Context Information in Pervasive Computing Systems, In: Pervasive '02: Proceedings of the First International Conference, 2. 2002, **Anais...** London, UK: Springer-Verlag, 2002, p.167-180.

HENRICKSEN, K. AND INDULSKA, J. Developing context-aware pervasive computing applications: Models and approach. **Journal of Pervasive and Mobile Computing (Elsevier)**, v. 2(1), p. 37-64, [S.l.], 2006.

HOBBS, J.; PAN, F. Time Ontology in OWL Homepage. 2006. Disponível em: <<http://www.w3.org/TR/owl-time/>>. Acesso em: 08 Maio 2008.

HORROCKS, I; PATEL-SCHNEIDER, P.; BOLEY, H.; MACGREGOR, S.; GROSOFF, B; DEAN, M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML HomePage. 2004. Disponível em: <<http://www.w3.org/Submission/SWRL/>>. Acesso em: 08 Maio 2008.

JINI Homepage. 2008. Disponível em: <<http://www.jini.org/>>. Acesso em: 08 Maio 2008.

KALLIO S.; KELA J.; MANTYJAARVI J.; PLOMP J. Visualization of hand gestures for pervasive computing environments. In: AVI '06: Proceedings of the working conference on Advanced visual interfaces, 2002, Venezia, Italy, **Anais...** New York, NY, USA: ACM, 2006, p. 480-483.

KRAUSE, A.; SMAILAGIC, A.; SIEWIOREK, D. Context-Aware Mobile Computing: Learning Context-Dependent Personal Preferences from a Wearable Sensor Array, **IEEE Transactions on Mobile Computing (IEEE Computer Society)**, v. 5(2), p. 113-127, Los Alamitos, CA, USA, 2006.

LEE, C.; HELAL, S.; LEE, W.. Universal Interactions with Smart Spaces. **IEEE Pervasive Computing (IEEE Educational Activities Department)**, v.5, p. 16, Piscataway, NJ, USA, 2006.

LING, J.; JIANYA, G.; BIN, L.; MIN, M.. GeoReferencing the Semantic Web Based on Geoontology. **Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on** p. 1545-1548, [S.l.], 2006.

LOPEZ, M.; GOMEZ-PEREZ, A.; SIERRA, J.; SIERRA, A. Building a chemical ontology using Methontology and the Ontology Design Environment. **Intelligent Systems and Their Applications, IEEE**, v. 14, 1, p. 37-46, [S.l.], 1999.

MANOLA, F.; MILLER, E. RDF Primer Homepage. 2004. Disponível em: <<http://www.w3.org/TR/rdf-primer/>>. Acesso em: 08 Maio 2008.

MARTIN, D.; BURSTEIN, M.; HOBBS, J.; LASSILA, O.; MCDERMOTT, D.; MCILRAITH, S.; NARAYANAN, S.; PAOLUCCI, M.; PARSIA, B.; PAYNE, T.; SIRIN, E.; SRINIVASAN, N.; SYCARA, K. OWL-S: Semantic Markup for Web Services. HomePage. 2004. Disponível em: <<http://www.w3.org/Submission/OWL-S/>>. Acesso em: 08 Maio 2008.

MCILRAITH, S.; MARTIN, D. Bringing Semantics to Web Services. **IEEE Intelligent Systems**, v. 18, p. 90-93, Piscataway, NJ, USA, 2003

MOKHTAR, S.; FOURNIER, D.; GEORGANTAS, N.; ISSARNY, V. Context-Aware Service Composition in Pervasive Computing Environments. In: Proceedings of the 2nd International Workshop on Rapid Integration of Software Engineering techniques (RISE'05), 2005, **Anais...**, [S.l.]: Springer Berlin / Heidelberg, 2005, p. 129-144.

MOKHTAR, S.; KAUL, A.; GEORGANTAS, N.; ISSARNY, V.. Towards Efficient Matching of Semantic Web Service Capabilities. In: Proceedings of WS-MaTe -

International Workshop on Web Services - Modeling and Testing, 2006, **Anais...**, Palermo, Italy, 2006, paginação irregular.

NOY, N.; MCGUINNESS, D. Ontology Development 101: A guide to Creating your First Ontology, (Technical Report, Stanford Medical Informatics), 2001.

OSGI Alliance Homepage. 2008. Disponível em: <<http://www.osgi.org/>>. Acesso em: 08 Maio 2008.

OWL API HomePage. 2008. Disponível em <<http://owlapi.sourceforge.net/>>. Acesso em: 08 Maio 2008.

OWL Web Ontology Language Use Cases and Requirements HomePage. 2004. Disponível em <<http://www.w3.org/TR/webont-req/>>. Acesso em: 25 Mar. 2008.

PATEL-SCHNEIDER, P.; HORROCKS. I. OWL 1.1 Web Ontology Language Overview Homepage. 2006. Disponível em: <<http://www.w3.org/Submission/owl11-overview/>>. Acesso em: 08 Maio 2008.

PELLET: The Open Source OWL DL Reasoner HomePage. 2008. Disponível em <<http://pellet.owldl.com/>>. Acesso em: 08 Maio 2008.

PINTO, H. S.; MARTINS, J. P.. Ontologies: How can They be Built? **Knowl. Inf. Syst. (Springer-Verlag New York, Inc.)**, v. 6, p. 441-464. New York, NY, USA, 2004.

PONCE, E.; BEINGOLEA, J.; ORRILLO, H. KOFUJI, S. Manipulación de Información de Contexto de una Casa Inteligente usando ontologías. In: V Congreso de Informática y Sistemas de Sudamérica COISIS 2006, 5. Peru, 2006, **Anais...**, Tacna, 2006, 2006, p. 1-8.

PONCE, E.; CASSIO V. PRAZERES, C. AND KOFUJI, S.; TEIXEIRA, C.; PIMENTEL, M.. SeCoAS: An Approach to Develop Semantic and Context-Aware Available Services, In: XIII Brazilian Symposium on Multimedia and the Web , 13. Rio Grande do Sul, Brazil, 2007, **Anais...**, Gramado, 2007, p. 182-189.

RAVERDY, P-G.; ARMAND, S.; ISSARNY, V. Scalability Study of the MUSDAC Platform for Service Discovery in B3G Networks. In: Proceedings of Wireless World Research Forum Meeting (WWRF-17), 2006, **Anais...**, Heidelberg, Germany, 2006, paginação irregular.

REI: A Policy Specification Language HomePage. 2005. Disponível em <<http://ebiquity.umbc.edu/project/html/id/34/Rei-A-Policy-Specification-Language>>. Acesso em: 25 Mar. 2008.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **Personal Communications, IEEE**, v. 8, p. 10-17, 2001.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on, 85-90, 1994, **Anais...**, Santa Cruz CA: IEEE Computer Society, 1994, p. 85-90.

SCHMIDT, A.; BEIGL, M.; GELLERSEN, H.-W. There is more to context than location. **Computers and Graphics (Elsevier)**, v. 23, p. 893-901, 1999.

SMITH, M.; WELTY, C.; MCGUINNESS D. OWL Web Ontology Language Guide HomePage. 2004. Disponível em: <<http://www.w3.org/TR/owl-guide/>>. Acesso em: 08 Maio 2008.

STRANG, T.; LINNHOF-POPIEN, C. A Context Modeling Survey, In: UbiComp4: Proceedings Workshop on Advanced Context Modelling, Reasoning and Management associated with the Sixth International Conference on Ubiquitous Computing, 2004, **Anais...**, Nottingham-England, sep. 2004, paginação irregular.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: principles and methods. **Data Knowl. Eng. (Elsevier Science Publishers B. V.)**, v. 25, p. 161-197, Amsterdam, The Netherlands, 1998.

SWRLTab Homepage. 2007. Disponível em: <<http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab>>. Acesso em: 08 Maio 2008.

THE OWL-S Editor Homepage. 2004. Disponível em: <<http://owlseditor.semwebcentral.org/>>. Acesso em: 08 Maio 2008.

THE PROTÉGÉ Ontology Editor and Knowledge Acquisition System Homepage. 2008. Disponível em: <<http://protege.stanford.edu/>>. Acesso em: 08 Maio 2008

TJOA, A.; JOMSHOAA, A.; SHAYEGANFAR, F. WAGNER, R. Semantic Web challenges and new requirements. **Database and Expert Systems Applications**, 2005. Proceedings. Sixteenth International Workshop on, p. 1160-1163, 2005.

USCHOLD, M.; KING, M. Towards a Methodology for Building Ontologies. In: IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing, 1995, **Anais...**, Montreal: AAAI Press, 1995, paginação irregular.

WALTENEGUS, D. Dynamic Generation of Context Rules. **Lecture Notes in Computer Science (Springer-Verlag)**, v. 3996, p. 102-115, jun. 2006.

WEISER M. The computer for the 21st century. **Scientific American**, p. 933-940, San Francisco, CA, USA , 1995.

ZHOU, Y.; CAO, J.; RAYCHOUDHURY, V.; SIEBERT, J.; LU, J. A Middleware Support for Agent-Based Application Mobility in Pervasive Environments. In: ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops, 2007, **Anais...**, Washington, DC, USA: IEEE Computer Society, p. 9, 2007.

APÊNDICE A - ROTEIRO PARA A ELABORAÇÃO DE UM ESTUDO DE CASO

Neste apêndice, é descrito brevemente um roteiro para a elaboração de um Estudo de Caso, a construção de ontologias em Protégé e posteriormente a implementação em Java.

1. Construção das ontologias

Para a elaboração de um Estudo de Caso como foi apresentado no Capítulo 4, com o OMC, deve primeiramente ser construído as ontologias para a modelagem de contexto.

Na construção do modelo de contexto proposto OMC foi utilizado o Método de Desenvolvimento 101, o qual consta de sete passos desenvolvidos em um processo iterativo (NOY; MCGUINNESS, 2001).

Em seguida, serão detalhados os passos do Método de Desenvolvimento 101, considerando a utilização das ontologias para a modelagem de contexto (OMC). Também será apresentada a utilização do Protégé 4.0.

1.1 Determinar o domínio e o âmbito da ontologia

O âmbito das ontologias do OMC é para a modelagem de contexto de Ambientes Pervasivos. Se fosse outro domínio, o OMC não serviria muito.

1.2 Considerar o reuso de outras ontologias

O domínio do OMC é de uma Casa Inteligente, por sua divisão em dois níveis (Seção 3.2), ele pode ser reutilizado em outro domínio. Deve ser considerado reutilizar as ontologias do OMC, e outras ontologias na modelagem do Estudo de Caso.

1.3 Enumerar importantes termos na ontologia

Realizar uma lista dos termos importantes na ontologia. Consideremos a modelagem de pessoas de uma Casa Inteligente. Os termos devem incluir definições de uma pessoa, informações de contato, informações de relações entre pessoas entre outros. Na Tabela 7 são apresentados os termos da ontologia **HomePerson**, a qual permite a modelagem de pessoas de uma Casa Inteligente.

Tabela 7 - Termos para a modelagem de pessoas de uma Casa Inteligente

Nome do termo	Descrição
Nome	O nome da pessoa
Idade	A idade atual da pessoa
Telefone	Telefone de contato
Sexo	O sexo da pessoa
Pai	Seu pai
Mãe	Sua mãe
Irmão	Seus irmãos
Amigo	Seus amigos
Grupo de pessoas	Um conjunto de pessoas a ser modelado
Família	A família a qual a pessoa pertence

1.4 Definir as classes e hierarquias de classes

Depois de realizar uma lista dos termos importantes na ontologia são definidas as classes. Na Figura 19, é apresentada a hierarquia de classes da ontologia **HomePerson**. A classe **Group** descreve os grupos que uma pessoa pertence e tem como subclasse **Family**. A classe **Person** tem como subclasses **AdultPerson**, **Man**, **Parent**, **Resident**, **Woman**, e **YoungPerson**. A classe **YoungPerson** é complemento da classe **AdultPerson**.

Figura 19 - Listado das classes da ontologia **HomePerson**

1.5 Definir as propriedades das classes

Depois de definir as classes são definidas suas propriedades. As propriedades se dividem em propriedades tipo de objeto e tipo de dado.

Na Figura 20 são apresentadas as propriedades tipo de objeto da ontologia **HomePerson**, as quais são **resident**, **isMemberOf**, **knows**, **parentOf**, **neighborOf**, entre outros. As propriedades **parentOf** e **neighborOf** são sub-propriedade de **knows**.

Figura 20 - Listado das propriedades tipo de objeto da ontologia **HomePerson**

Na Figura 21 são apresentadas as propriedades tipo de dado, as quais são **lastName**, **organizationName**, **groupName**, **address**, **age**, **name**, entre outros. A propriedade **age** define a idade de uma pessoa, o tipo de dado desta propriedade é inteiro. A propriedade **name** define o nome de uma pessoa, o tipo de dado desta propriedade é *String*.

Figura 21 - Listado das classes da ontologia **HomePerson**

1.6 Definir a cardinalidade, tipo, domínio e valor das propriedades

Depois de definir as propriedades é definida a cardinalidade, tipo, domínio e valor das propriedades. A propriedade **hasSex** é funcional já que uma pessoa somente pode ter uma pessoa. O domínio desta propriedade é a classe **Person** e seu valor é a classe **Sex**.

1.7 Definir as classes e hierarquias de classes

Depois de definir as classes e propriedades são definidas as instâncias da ontologia. Na Figura 22, apresenta-se uma lista das instâncias da ontologia **HomePerson**. **Person_1**, **Person_2**, **Person_3**, **Person_4**, **Person_5** e **Person_6** são instâncias da classe **Person**, e representam as pessoas. **Male** e **FemaleSex** representam os gêneros masculino e feminino respectivamente.

Figura 22 - Listado das instancias da ontologia *HomePerson*

2. Implementação em Java

Para a manipulação dos dados utilizando dispositivos reais e sua integração com a modelagem de contexto, devem ser construídas interfaces implementadas em Java. Em seguida, serão definidos os passos para implementar o Estudo de Caso.

Inicialmente devem ser importadas as bibliotecas Java necessárias, posteriormente criar uma pasta que contém as ontologias do OMC criadas com o Protégé, e finalmente o código Java de implementação.

2.1 Definir as classes e hierarquias de classes

Para implementar em Java o Estudo de Caso devem ser importadas as bibliotecas OWL-API e Pellet. A versão utilizada do OWL-API é 2.1.1 a versão do Pellet é 1.5.1. Na Figura 23, apresenta-se o projeto OMC_Demo mostrando as bibliotecas necessárias.

Figura 23 - Livrarias necessárias para a implementação do Estudo de Caso

2.2 Criar uma pasta com as ontologias do OMC

Posteriormente tem que ser criado uma pasta com as ontologias do OMC. Na Figura 24, é apresentado a pasta que contém as ontologias do OMC, no projeto OMC_Demo.

Figura 24 - Pasta das ontologias do OMC para a implementação do Estudo de Caso

2.3 Inferência semântica do OMC em Java

Neste ponto para realizar a inferência semântica do OMC em Java são realizados dois passos, no primeiro mostra-se como carregar uma ontologia, criar um gerenciador e a máquina de inferência Pellet. No segundo a realização da inferência

No Quadro 23, apresenta-se parte de código em Java para carregar uma ontologia com o OWL-API e criar a máquina de inferência Pellet.

- é definida uma variável base, que contém o *Namespace* da ontologia (linha 0);
- é definida uma variável (*localPath*) (linha 1), que contém o endereço das pastas com as ontologias (Figura 24);
- é criado um gerenciador (*manager*) para a manipulação das ontologias (linha 2);
- é criada uma nova ontologia (linha 3);
- é utilizado o gerenciador para carregar a ontologia (linhas 6-7);
- é criada uma máquina de inferência, neste caso utiliza-se o Pellet (linhas 8-9)

```

0   String base = "http://www.pad.lsi.usp.br/context/
      homePerson_data1.owl";
1   String localPath = "file:/D:/_Docs/ontFile/ontologies2/";
2   OWLOntologyManager manager = null;
3   URI physURI_DAT1 = URI.create(localPath + "homePerson_data1.owl");
4   OWLOntology ont_DAT1 = null;
5
6   manager = OWLManager.createOWLOntologyManager();
7   ont_DAT1 = manager.loadOntologyFromPhysicalURI(physURI_DAT1);
8   Reasoner reasoner = new Reasoner( manager );
9   reasoner.loadOntology(ont_DAT1);
...

```

Quadro 23 - Parte de código em Java para carregar uma ontologia

No Quadro 24, apresenta-se parte de código em Java para realizar inferência utilizando a máquina de inferência Pellet. Depois de ser criada a máquina de inferência como é definido no Quadro 23, se verifica se a ontologia é consistente (linha 0), se for consistente se passa a realizar a inferência, em caso contrario se apresenta uma mensagem de que a ontologia é inconsistente (linhas 7-11). Se a ontologia for consistente é aplicado a inferência (linhas 4-5)

```

0   if(reasoner.getKB().isConsistent())
1   {
2       System.out.println("");
3       System.out.println("Consistente");
4
5       reasoner.getKB().realize();
6       reasoner.getKB().classify();
7   }
8   else
9   {
10      System.out.println("");
11      System.out.println("Inconsistente");
12  }

```

Quadro 24 - Parte de código em Java para realizar inferência